Improving Routing in Networks of UAVs via Scoped Flooding and Mobility Prediction

Yi Li
School of Electronics and Information
Northwestern Polytechnical University
Xi’an, China
npuliyi@gmail.com

Marc St-Hilaire and Thomas Kunz
Department of Systems and Computer Engineering
Carleton University
Ottawa, Canada
marc_st_hilaire@carleton.ca, tkunz@sce.carleton.ca

Abstract—**Due to their specific characteristics, Unmanned Aeronautical Ad-hoc Networks (UAANETs) can be classified as a special kind of mobile ad-hoc networks. Due to the high mobility of Unmanned Aerial Vehicles (UAVs), designing a good routing protocol for UAANETs is challenging. Recently, a new protocol called Reactive-Greedy-Reactive (RGR) [1] has been proposed as a promising routing protocol in high mobility and density-variable scenarios. Although the RGR protocol improves the packet delivery ratio, the overhead and delay are higher when compared to AODV [1]. In this paper, a scoped flooding and mobility prediction based RGR protocol is proposed to improve the performance of RGR in UAANETs. Simulation results show that the new protocol can effectively enhance the performance of the RGR protocol in terms of packet delivery ratio, overhead, and delay.**

Keywords—Unmanned Aeronautical Ad-hoc Networks (UAANETs); Unmanned Aerial Vehicles (UAVs); Routing Protocol; Reactive-Greedy-Reactive (RGR) protocol; Mobility Prediction; Scoped Flooding

I. INTRODUCTION

An Unmanned Aerial Vehicle (UAV), commonly known as a drone, is an aircraft with no human operator on board. In many applications, several UAVs (referred to as a flock) must cooperate with each other to reduce mission delay and increase reliability in critical aerial operations [2]. These UAVs form an ad-hoc network in the specific area and have the ability to share information with others. This kind of ad-hoc network is referred to as Unmanned Aeronautical Ad-hoc Network (UAANET) [3]. UAANETs have unique characteristics such as high mobility and relatively low number of UAVs in the network. These properties can cause constant topology changes due to numerous link breaks in the network. Therefore, UAANETs require an efficient routing protocol to minimize the effect of these limiting topological features. Although many traditional routing protocols have been proposed for Mobile Ad-hoc Networks (MANETs) [4, 5, 6], they cannot be directly applied to UAANETs due to their unique features. As a result, it is necessary to develop new efficient routing protocols that appropriately address these limiting topological features and exploit UAV-specific characteristics. Recently, a new protocol, called Reactive-Greedy-Reactive (RGR) [1], which combines the advantages of reactive routing and Greedy Geographic Forwarding (GGF) [7] has been proposed for high mobility and sparse scenarios. The protocol exploits the fact that UAVs have access to accurate location information for navigation purposes. At the same time, it avoids the need of an independent location service by integrating the propagation of location information into the reactive routing protocol. However, one of the main drawbacks of this protocol lies in the lack of regularly updated information regarding the location of the next hops. Indeed, due to the high mobility of the nodes, data packets could be dropped if they are sent to an outdated location. Another drawback exists in that the protocol overhead is still relatively high due to flooding of route request messages.

In this paper, we suggest a scoped flooding and mobility prediction based RGR to enhance the performance of the original RGR protocol. In addition to the location information, this mechanism takes advantage of the velocity vector of the nodes to predict their current locations. Unlike many other mobile nodes, the trajectories of UAVs are less prone to abrupt changes, so we expect this to lead to very good prediction accuracy. Mobility prediction then allows the protocol to monitor the status of the reactive routes and select appropriate neighbors during the GGF phase of the protocol. At the same time, two different scoped flooding methods are utilized to reduce the overhead messages generated by the original RGR protocol during the route discovery phase by exploiting location information.

The rest of this paper is organized as follows. Section II reviews the RGR protocol, scoped flooding based protocols, and mobility prediction mechanisms. In Section III, the proposed scoped flooding and mobility prediction based RGR is introduced and described. Section IV presents the simulation settings, followed by the simulation results in Section V. Finally, we conclude in Section VI.

II. OVERVIEW OF RELATED PROTOCOLS

A. Reactive-Greedy-Reactive (RGR) Protocol

The basic idea behind RGR, which has been proposed in [1], is to combine a reactive protocol (in this case AODV) with greedy geographic forwarding. In this protocol, if there is no valid route for data packets to be transmitted, the source node of the data packets begins a route discovery process (as in AODV) in order to find a valid route entry to reach the destination node, flooding Route Request (RREQ) packets into the network. In fact, a reactive route is established when the source node receives the Route Response (RREP) packet from the destination node. Once the route is established, data packets

978-1-4673-4404-3/12/$31.00 ©2012 IEEE

buffered at the source can be transmitted to the destination. The novelty in RGR is that the location information of the destination node is obtained by every intermediate node as the RREP packet propagates back to the source node. In the route maintenance process, if an intermediate node cannot receive 3 successive hello messages, the link is considered lost and the reactive route breaks. RGR invalidates the reactive route and switches to the GGF mode. In this mode, the protocol sends the data packets to the neighbor node which is closest to the destination node (in essence, salvaging it). At the same time, a Route Error (RERR) packet will be sent back to the precursor node until it reaches the source node. The source node, if it has more data to transmit, initiates a new route discovery process in order to establish a new reactive route to the destination. Packet forwarding via GGF will continue till the data packet reaches the destination node, is dropped by an intermediate node due to the TTL parameter reaching to 0, or greedy forwarding fails to find a neighbor node closer to the destination. Similar to other geographic routing protocols, RGR keeps track of each neighbor's existence and location by having nodes periodically broadcast hello messages once every second that contain a node's ID and location information.

B. *Scoped Flooding Based Protocol*

LAR [8] is one of the most popular proposals to reduce overhead messages in AODV and DSR [5, 6]. Because the geographic information of the destination is not directly available in AODV or DSR, LAR utilizes the original AODV or DSR protocol to establish connectivity with the destination node. During this phase, the source node will learn the geographic information of the destination from the route reply message sent by the destination node or by an intermediate node which knows the latest route to the destination. Based on this location information, LAR does not need to flood the route request packet into the whole network. It confines the flooding of RREQ packets to the part of the network that approximately contains the destination node. During the route discovery process, every intermediate node will compare its own location information with the specified search area contained in the RREQ packet. If it belongs to the search area, this node will rebroadcast the RREQ packet. Otherwise, the RREQ packet will be discarded [9]. When only paths outside the LAR search area are able to reach the destination, LAR will fall back to flooding the RREQs. Under LAR, the geographic information is only utilized to scope the region of the route request message propagation and not used to decide how to forward data packets.

C. *Mobility Prediction Based Protocol*

Different versions of on-demand routing protocol based on mobility prediction [10, 11, 12, 13] have already been proposed. Most of these protocols focus on selecting the most stable route from already known backward routes. Those backward routes are set up once the destination node receives the RREQ messages from different neighbor nodes. In [10], Link Expiration Time (LET) between any mobile nodes has been exploited to improve various unicast and multicast routing protocols. By piggybacking location information on control packets, the link expiration time is estimated between any two nodes and appended to the RREQ message. The intermediate node will broadcast this RREQ message to all neighbors. When receiving RREQ messages, the destination node will learn the LETs of all known links and decide which link has the maximum Route Expiration Time (RET), which is defined as the least of the LET values of one link. Using RET, a more stable route can be set up for data transmission.

The MPRP protocol [11] is proposed to predict link status during the data transfer phase. In this protocol, location information is included in the data packet. During data transmission, an intermediate node can extract the location information of the previous node from the data packet. By comparing the distance difference between two consecutive received data packets, this current node has two functions: i) judge when the link will break, and ii) find out unnecessary nodes on the route which are too close to the current node. The closest node should be replaced with a two hop node as new next hop. This mobile prediction method is simple and does not require complicated computation and beacon packets. However, this mechanism must add a prediction table to the on-demand protocol and makes use of a new message called Route Expired (REXP) message to feedback the link status to a previous node. In our work, we added a similar capability to RGR, yet avoided the need for a new protocol message.

III. SCOPED FLOODING AND MOBILITY PREDICTION IN RGR

A. *Scoped Flooding in RGR*

The original RGR protocol inherited RREQ flooding to the whole network during route discovery process from AODV, optionally using an expanded ring search technique. We call this strategy blind flooding in the remainder of the paper. Although the number of UAVs in the network is relatively small, blind flooding adds high protocol overhead, potentially resulting in buffer overflow and network congestion. In order to reduce the number of RREQ packets, this section discusses two different mechanisms of scoped flooding in RGR.

The first mechanism is as follows. When a route discovery process is initiated for the first time, the source node floods the RREQ packets into the whole network and waits for the RREPs from the destination node. When the RREP packets arrive at the source node, a valid reactive route will be set up and, in the meantime, the location information of the destination node will be learnt by the source node. After a short period of time, a new route discovery process may need to be performed for the same destination node due to a route break caused by the highly dynamic topology of our UAANET scenarios. In this case, using the geographic information of the destination learnt previously, the source node calculates the distance to reach the destination and includes this result in the RREQ packet (as well as its knowledge of the destination's location). This new request packet is broadcast to all neighboring nodes. Upon receiving the RREQ packet, a neighbor node extracts the distance value from the RREQ packet and recalculates its own distance to reach the destination node. If this new distance is less than the distance from the RREQ packet, the neighbor node should replace the old value with the new one in the RREQ packet and rebroadcast the packet to its neighbors. Otherwise, this RREQ packet will be discarded. This process continues until the RREQ packet reaches the destination node, which then replies via a RREP, updating its location information in the process. A source node will wait to receive a route reply to the scoped RREQ. If the geographic information is out of date, this scoped flooding may fail and the source will issue another RREQ after a predefined timeout, increasing the source-destination distance by a fixed percentage. In our implementation, we used an increase of 20% for each repeated RREQ. The RREQ carries a repetition counter, allowing intermediate nodes to similarly apply an increased distance to the destination with each repetition. In essence, this provides some additional "slack" in the RREQ propagation. After a specific number of retries, say 5, the source node will switch from scoped flooding to blind flooding.

The second mechanism depends on the facts that not only the source node but also other nodes in the network learn the destination location in RGR. When route discovery is initiated the first time, the source node will set the distance to destination to zero and adds this to the RREQ packet. Thereafter, the source node broadcasts the RREQ packet to all neighbors. Every neighbor receiving the RREQ packet first checks whether it has geographic information related to the destination node. When a node does not know the destination location, it rebroadcasts the RREQ packet. Otherwise, the intermediate node calculates its own distance to the destination node and compares it with the distance value in the RREQ packet. If the distance value extracted from the RREQ is zero (to say that the previous node does not know the destination location), the intermediate node includes the calculated distance into the RREQ packet and rebroadcast it. If the distance value extracted from the RREQ is nonzero, the intermediate node compares this distance value to its own distance to the destination as above. If the node's distance is less than the distance value from the RREQ, the RREQ distance value will be updated and the RREQ rebroadcast. Otherwise, the intermediate node drops the RREQ packet. This process is repeated until the RREQ packet reaches the destination. Assuming that no node in the whole network knows the geographic location of the destination, this mechanism degrades to blind flooding. On the other hand, unlike the first idea, we do not necessarily need to resort to blind flooding the first time a route request is issued. If a source node uses inaccurate location information, this version of scoped flooding may fail as well. In this case, a source will re-issue a RREQ with 0 distance after unsuccessfully waiting for a RREQ.

B. *Mobility Prediction in RGR*

According to the RGR protocol, data packets are sent to the destination node once a reactive route is established. During transmission, the intermediate nodes detect the status of the next hop by receiving hello messages. If an intermediate node fails to obtain 3 consecutive hello messages from the next hop, this intermediate node will then conclude that the link to reach the next hop is broken. At this time, data will be alternatively forwarded by the GGF mechanism. Given that hello messages are broadcast once every second, this mechanism delays link break discovery by between 2 to 3 seconds. When a link break takes place, the intermediate node cannot access the current link status immediately and has to wait (in the worst case up to 3 seconds or 3 hello intervals) before it can act on it. During this time, the intermediate node still assumes the link is valid and continues to forward data packets through the (falsely) existing reactive route. As a result, these data packets will be lost and cannot be salvaged by the GGF mechanism. Note that we could set the criteria for link breakage to a different number, such as a single missed hello message. While this would reduce the time it takes to detect an actual link break, it would also lead to many incorrect RERR messages, as hello messages, being broadcast in the wireless media, could get lost due to interference or collisions. Alternatively, we could reduce the hello interval, but with every node periodically transmitting hello messages, this would increase the protocol message overheads significantly. In order to solve this problem, the proposed mobility prediction mechanism employs the velocity vector, which is associated with a timestamp, of the next hop node to compute the distance between the current node and the next hop node before forwarding data packets (which is part of the periodic hello message). As soon as the next hop node is out of transmission range, the current transmitting node can immediately respond by invalidating the status of the reactive route and, at the same time, switch to GGF to salvage the data packets that would have been dropped otherwise.

The functionality and propagation of control messages are similar to RGR except that RREQ, RREP, and hello messages carry more information. In the route discovery process, RREQ and RREP messages carry not only the location information of the destination node, but also the speed, direction, and timestamp of the precursor node. In the route maintenance process, hello messages periodically broadcast information including location, speed, direction, and timestamp to neighbors. These parameters are extracted from these messages and recorded in every intermediate node.

After the route discovery process, the source node sends buffered data packets to the destination node. Every intermediate node relays data packets one by one. Unlike RGR, the current node, which is about to transmit the packet, first checks the distance to the next node. With the help of (1), the current node estimates the real-time position of the next hop (note that for simplicity, we express this in a 2D coordinate system, but it would be relatively straightforward, though more involved, to express these relations in a 3D coordinate system as well).

$$\begin{aligned} X_{predict} &= X_{next} + V \times \cos(\theta) \times (current\_time - timestamp) \\ Y_{predict} &= Y_{next} + V \times \sin(\theta) \times (current\_time - timestamp) \end{aligned} \tag{1}$$

In (1), $X_{predict}$ and $Y_{predict}$ are the X and Y coordinates of the predicted location of the next hop. X_{next} and Y_{next} are the last known location of the next node. The timestamp records the time at which the last known location was recorded. Parameters V and θ represent the speed and the direction of the next hop node respectively. These necessary parameters are extracted from the routing table maintained in the current node.

Using (2), the current node judges whether the next hop is out of transmission in real-time.

$$D_{next} = \sqrt{(X_{own} - X_{predict})^2 + (Y_{own} - Y_{predict})^2} \tag{2}$$

In (2), D_{next} is the actual distance from the current node to the next hop node. X_{own} and Y_{own} are the current node's location. $X_{predict}$ and $Y_{predict}$ are the X and Y coordinates obtained from (1).

If D_{next} is smaller than the transmission range, the current node continues to transmit data packets to the next hop using the reactive route. However, if the distance is greater than the transmission range (i.e. the next hop in the reactive route is now out of range), the current node immediately stops sending data packets over this route and simultaneously switches to GGF to forward data packets. During the GGF phase, the current node obtains the real-time topology of neighbors by exploiting the same mobility prediction method. It then selects the node closest to the destination to greedy forward packets towards it.

Note that mobility prediction requires nodes to be at least approximately time-synchronized. If we assume a maximum travel speed of 300 km/h (or equivalently 83.3 m/s) for a UAV, clock synchronization errors of 1ms translate into an error in the predicted location of at most 8.3 cm if two UAVs travel in opposite directions, a very small fraction of typical transmission ranges. Such synchronization accuracy is easily achieved with one of the many clock synchronization protocols proposed in the literature [14, 15]. In addition, if UAVs obtain their location information for navigation purposes via GPS, all nodes will also be synchronized tightly to a very accurate global reference time, obliviating the need for a separate clock synchronization protocol.

Since GGF is used as a fallback mechanism, the RERR packet does not have to be generated immediately when an intermediate node detects that the link to the next hop node is broken. So we delay the transmission of a RERR message after detecting a link break by 3.5 seconds, a value that exceeds the longest delay for AODV and original RGR to detect a link break (3 seconds or alternatively 3 consecutive hello intervals). A purely reactive routing protocol such as AODV has to re-establish a new route as soon as possible, to prevent long gaps in data packet transmissions. During this period, in RGR, data packets can be salvaged via GGF. In fact, a previous study showed that for a small number of hops, GGF has a high success probability to reach the destination [16]. Until a new reactive route is established, an intermediate node can keep on sending data packets to a neighbor node which is closest to the destination node. In highly dynamic topologies, by carefully selecting an appropriate RERR message delay, we expect this to reduce the total number of RREQs initiated by a source without impacting overall protocol performance.

IV. SIMULATION SETTTINGS

In order to evaluate the performance of RGR with the proposed enhancements, we set up a specific scenario via OPNET Modeler 16.0 [17]. 15 UAVs are distributed randomly in the initial region. A free path loss propagation model is considered in the simulation. Every UAV in the scenario randomly selects another UAV as a target to send data packets. The packet sizes are drawn from an exponential distribution with mean 1024 bits. The inter-packet delays follow an exponential distribution with mean 0.2 seconds. The reason for this traffic flow structure is that the adaptability of the proposed modified RGR protocol in processing multi-flows can be tested. The capability to handle multi-flows is an important characteristic of a routing protocol [18]. Thus, this scenario can be considered as a relatively realistic multi-traffic flows scenario for UAANETs.

TABLE I. MOBILITY PARAMETERS

Parameters	***Values***
Speed	Uniform(50,60) m/s
Initial Region	1×1 km^2
Search Size	2×4 km^2
Number of UAVs	15
Transmission Range	1,000 m
Simulation Time	1,000 sec

The Random Waypoint (RWP) model is used to simulate realistic UAV mobility for a search mission. In a search mission, every UAV is looking for an object in a specific area. Because each UAV must move continuously without pause, the pause time in the model for every node is set to 0. In order to simulate a high mobility scenario, we choose the speed for every UAV to be uniformly distributed between 50 and 60 m/s [19]. We do not believe that the RWP model is a proper description of UAV mobility, as it allows for very abrupt and sharp changes in a UAV's trajectory, but have not yet completed work on more realistic scenarios. As discussed above, more realistic UAV trajectories are also more predictable. Therefore the results presented here underestimate the performance gains that mobility prediction is able to achieve. The transmission range of each UAV is set to 1,000 meters and the simulation time is set to 1,000 seconds. In the initial phase, all UAVs are in each other's vicinity. So, there will be good initial networking performance independent of any routing protocol. As the UAVs gradually spread over the search region during the simulation, the performance in terms of packet delivery ratio, overhead, and delay will deteriorate and eventually reach a steady-state behavior. The specific mobility parameters are listed in Table I.

V. SIMULATION RESULTS

For the simulations, 10 different seed values of the pseudo random number generator are set in OPNET, so that each set of simulation results will be independent. Five protocols, including AODV, original RGR, RGR with mobility prediction (MPRGR), RGR with scoped flooding method 1 and mobility prediction (SF1MPRGR), and RGR with scoped flooding method 2 and mobility prediction (SF2MPRGR) are simulated individually and compared with each other.

In Figures 1, 2 and 3, the performance of the above five protocols is evaluated in terms of packet delivery ratio (PDR), protocol overhead (measured in control packets transmitted per second), and packet end-to-end delay. As can be seen from Figure 1, MPRGR has the highest PDR among the five protocols, reaching approximately 83%. The two scoped flooding based protocols have almost similar results, with very little degradation compared to MPRGR. Meanwhile, the original RGR and AODV perform much worse than the other three protocols. Their PDR performance drops to 80% and 76%

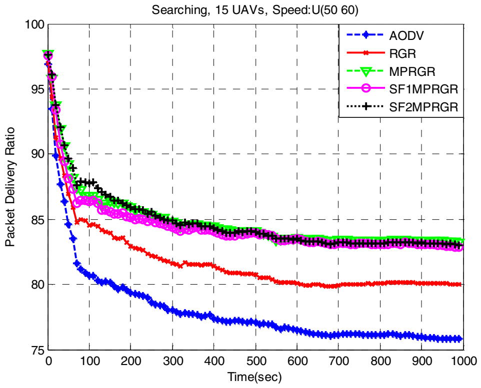

Figure 1. Packet delivery ratio

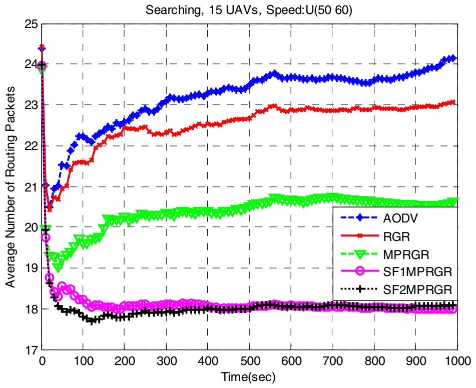

Figure 2. Average routing traffic

respectively in steady state. The main reason to explain this difference is that both original RGR and AODV do not have the capacity to check the link status on a reactive route in real-time. Detecting the status of a reactive route is delayed by up to 3 hello intervals. The other three protocols, on the other hand, have the ability to detect the status of the reactive route during data transmission and switch to GGF as soon as a link break takes place. Therefore, packets that are dropped by the original RGR and AODV (as they are transmitted over invalid links) are salvaged by the other three protocols.

From Figure 2, we can see that the two scoped flooding protocols have the lowest protocol overhead, reducing the overhead of MPRGR from almost 21 packets per second to 18 packets per second. The results verify that both scoped flooding mechanisms, by exploiting geographic information successfully, reduce the amount of RREQs during the simulation. The overhead of MPRGR is about 3 packets per second lower than the original RGR and about 4 packets/second for AODV. MPRGR is waiting 3.5 seconds before sending a RERR back to the source thus reducing the total number of RREQ. Consequently, the number of RREQs initiated by the source node is decreased resulting in a reduced overhead.

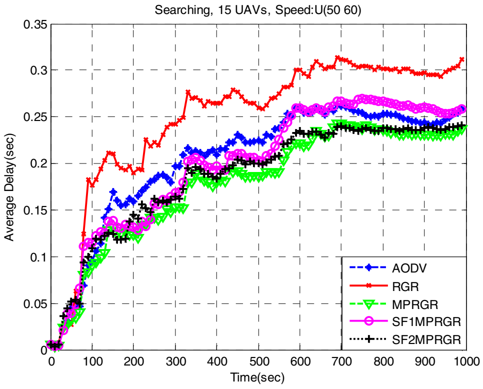

Figure 3. Average packet delay

In terms of end-to-end delay, we can see from Figure 3 that the delay for the original RGR is high compared to the other protocols. The original RGR's delay is about 300ms. Meanwhile, the other four protocols have similar average delay (approximately 250ms) in steady state. This shows that the improvements in PDR and control message overhead do not come at a cost with respect to delay. Certainly the reduction in control messages, leading to less overall network traffic, benefits the three new proposed protocol variants. In addition, with delay calculated over all packets that are received, the delay calculated for AODV is somewhat misleading: only those packets that are delivered over the reactive route will be considered. Salvaging packets in general will result in those packets being delivered over more hops, increasing their end-to-end delay. But this (relatively small) incremental latency is a small price to pay for a substantial increase in PDR.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, a scoped flooding and mobility prediction based RGR protocol is proposed. The new protocol takes advantage of the location information and velocity vector of nodes to assess the real-time status of the next hop node. Based on the status of the reactive route, every intermediate node has the ability to decide whether to send data packets through the reactive route or switch to GGF immediately. This modification improves PDR and reduces control message overhead. In order to further reduce the protocol control message overhead, two scoped flooding mechanisms are applied and combined with MPRGR, called SF1MPRGR and SF2MPRGR respectively.

Our simulation studies show that scoped flooding and mobility prediction result in significantly higher packet delivery ratio, lower overhead, and lower end-to-end delay compared to the original RGR and AODV protocols. From these results, we can conclude that it is critical to check the real-time status of the next hop node during the data transfer phase and both scoped flooding mechanisms are effective in suppressing the flooding of RREQ control messages.

For future work, we will explore mobility prediction further. For example, we have not yet systematically studied the prediction errors. As UAVs change direction, past trajectory information will not always allow us to accurately predict current UAV locations. This is in particular a problem as location and velocity information gets older. As part of this effort, we also need to conclude work on realistic mobility trajectories for UAVs. We also plan further modifications to improve the proposed protocol. For example, in the GGF phase, the criterion to select the next hop node should not only be based on the closest distance to reach the destination node. It could also include additional parameters, such as link stability, link data rate, etc. Similarly, we could piggyback path stability and location information of the UAVs onto data packets to allow for more frequent updates or use information about link stability in determining more stable reactive paths (the current criteria, again inherited from AODV, is to select the path with the minimum number of hops, which, all else being equal, results in paths over relatively long and brittle links). Finally, we have not explored the differences in the two scoped flooding approaches further. As shown by our results, the two approaches perform comparable, so we plan to conduct further studies with a range of different scenarios to understand the relative strengths and weaknesses of each approach better.

ACKNOWLEDGMENT

The authors would like to thank Rostam Shirani, Md. Arifur Rahman, Muhammad Raisul Alam for continuous and helpful discussions.

REFERENCES

[1] R. Shirani, M. St-Hilaire, T. Kunz, Y. Zhou, J. Li, and L. Lamont, "Combined reactive-geographic routing for unmanned aeronautical ad-hoc networks," in *Proceedings of the 8th International Wireless Communications and Mobile Computing Conference* (IWCMC 2012), Aug. 2012.

[2] E. Frew and T. Brown, “Airborne communication networks for small unmanned aircraft systems,” *Proceedings of the IEEE*, vol. 96, no. 12, pp. 2008–2027, Dec. 2008.

[3] R. Shirani, M. St-Hilaire, T. Kunz, Y. Zhou, J. Li, and L. Lamont, “Quadratic estimation of success probability of greedy geographic forwarding in unmanned aeronautical ad-hoc networks,” in *Vehicular Technology Conference Fall (VTC 2012-Spring), 2012 IEEE 75th*, May. 2012.

[4] T. Clausen and P. Jacquet, “Optimized Link State Routing Protocol RFC3626,” http://www.ietf.org/rfc/rfc3626.txt.

[5] E. Perkins, C. Belding-Royer and S. Das, “Ad hoc On-Demand Distance Vector (AODV) Routing RFC3561,” http://www.ietf.org/rfc/rfc3561.txt.

[6] D. Johnson et al, “Dynamic Source Routing in Ad Hoc Wireless Networks RFC4728,” http://www.ietf.org/rfc/rfc4728.txt.

[7] A. Nayak and I. Stojmenovic, “*Wireless Sensor and Actuator Networks: Algorithms and Protocols for Scalable Coordination and Data Communication,”* Wiley, 2010.

[8] Y. Ko and N.H. Vaidya, “Location-Aided Routing (LAR) Mobile Ad Hoc Networks”, in Proc. ACM/IEEE Mobicom, pp. 307-321, Oct. 1998.

[9] V. Hnatyshin, M. Ahmed, R. Cocco, and D. Urbano, “A Comparative Study of Location Aided Routing Protocols for MANET,” in Wireless Days (WD), pp. 1-3 Oct. 2011.

[10] N.-C. Wang and S.-W. Chang, “A reliable on-demand routing protocol for mobile ad hoc networks with mobility prediction,” Computer Communications, vol. 29, no. 1, pp. 123-135, Dec. 2005.

[11] M. Hasanlou and H. Mohammadi, “MPRP: A novel mobility prediction algorithm for improving routing protocols of mobile ad hoc networks,” in Iranian Conference on Electrical Engineering (ICEE2008).

[12] B.H. Sathyaraj and R.C. Doss, "Route maintenance using mobility prediction for mobile ad hoc networks," *Mobile Adhoc and Sensor Systems Conference, 2005. IEEE International Conference on* , vol., no., pp.6 pp.-101, 7-7 Nov. 2005.

[13] L.M. Meng, J.X. Zang and W.H. Fu, “A Novel Ad hoc Routing Protocol Research Based on Mobility Prediction Algorithm,” *Wireless Communications, Networking and Mobile Computing, 2005. Proceedings. 2005 International Conference on* , vol.2, no., pp. 791-794, 23-26 Sept. 2005.

[14] Y.-Chung Wu, Q. Chaudhari and E. Serpedin, "Clock Synchronization of Wireless Sensor Networks," *Signal Processing Magazine, IEEE*, vol.28, no.1, pp.124-138, Jan. 2011.

[15] B. J. Choi, H. Liang, X. M. Shen and W. H. Zhuang, "DCS: Distributed Asynchronous Clock Synchronization in Delay Tolerant Networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol.23, no.3, pp.491-504, March 2012.

[16] R. Shirani, M. St-Hilaire, T. Kunz, Y. Zhou, J. Li, and L. Lamont, “The Performance of Greedy Geographic Forwarding in Unmanned Aeronautical Ad-hoc Networks,” in *9th Annual Conference on Communication Networks and Services Research Conference* (CNSR 2011), pp. 161-166, May 2011.

[17] “OPNET, Application and Network Performance,” http://www.opnet.com.

[18] E. Frew and T. Brown, “Airborne communication networks for small unmanned aircraft systems,” *Proceedings of the IEEE*, vol. 96, no. 12, pp. 2008–2027, Dec. 2008.

[19] “Unmanned aerial vehicles classification,” Available at: http://www.vectorsite.net/twuav.html.