The Reliable Routing of the Air-Ground Integrated Network based on Kalman Filter Prediction

Zhimin Zhang
School of EECE
University of Chinese Academy of Sciences
Beijing, China
zhangzhimin212@mails.ucas.ac.cn

Fei Qin
School of EECE
University of Chinese Academy of Sciences
Beijing, China
fqin1982@ucas.ac.cn

***Abstract*—The Air-Ground integrated networks are consisted with both stationary ground-based nodes and fast-moving air-based nodes. Nonetheless, the topology will be fast changed caused by the fast movement of the aircraft. The classical AODV algorithm may not suit this scenario and may suffer from the unstable transmission, large delay, and packet loss. This paper propose a Kalman filter aided AODV routing protocol to increase the performance of air-ground integrated networks. The Kalman filter has been employed to predict the link quality of air node with linear model in short time scale, which will be able to form a proactive routing framework. The proposed trajectory prediction method is simple to implement and can effectively solve the problem of fast topology change due to the movement of aircraft. The dynamic repair package is added to the route repair strategy to provide a reliable auxiliary solution to reducing the route interruption rate and the number of transmission links created, which will finally improve the link stability.**

***Index Terms*—AODV Routing Protocol, Link Quality Prediction, Kalman filter, Air-Ground network**

I. INTRODUCTION

The Ad hoc On-Demand Distance Vector (AODV) routing protocols have been widely utilized in Ad Hoc networks, where the route discovery and maintenance are based on the established routing links [1]. Nonetheless, any movement of network nodes will have an impact on the transmitting data. The AODV routing protocol copes with topology change at the cost of delayed routing, as it only tries to find the best route upon the arrival of traffic [2]. This pattern works for most Ad Hoc scenarios, but may not be affordable in the air-ground network, where the network is contributed by both the static or slow moving ground node and the fast moving air node. In this scenario, the topology will be changed regularly, and the the original shortcomings of AODV algorithm will be significantly amplified. This will have a large impact on the link quality, and the real-time of data transmission can not be satisfied due to excessive transmission delay. Without any doubt, the classical route discovery strategy of AODV can be improved by taking the predicted node location into account to re-arrange the path shortest algorithm, which can effectively avoid the frequent route discovery search due to fast node movement and also extend the link life cycle. Therefore, the performance of route discovery can be significantly improved.

In this context, the classical AODV route discovery strategy should be improved through the incorporation with prediction algorithms, e.g. the Kalman filter [3]. The Kalman filter can predict node positions and movement direction based on the motion state of node, which models the motion activity into a state space with linear fitting. On this basis, the direction of motion of the source node can be predicted in the process of route discovery, and the direction of node motion is utilized as a prerequisite for establishing the route. Benefited by this, the node movement will have less impact on the established route, since the route does not need to be established frequently and the intermediate nodes do not need to flood frequently to find the target node. As a result, the delay of network data transmission can be decreased while the energy consumption can be reduced.

The rest of this paper will be organized as follows: section II will provide the methodology, section III will validate the performance of proposed Kalman filter aided with extensive simulations, and finally the conclusion and future works will be discussed in section IV.

II. METHODOLOGY

A. Route discovery protocol design based on node motion direction prediction

The space-based self-assembled network usually consists of numerous high-speed mobile nodes, and the actions of each node are executed according to the arranged missions. However, due to environmental conditions and unexpected situations, the trajectory could be varied from the plan. For this reason, the Kalman filter can be applied to predict the future flight information with the current known positions, which has been widely utilized for the prediction of node position [4]. Suppose the position coordinates of the aircraft at moment k are $(x(k), y(k), h(k))$ and the velocities in the X,Y,Z-axis directions are $x'(k), y'(k), h'(k)$, respectively, which are the derivatives of the displacement increments in the X,Y,Z-axis directions with respect to time k. Then we have:

$$\begin{cases} x(k) = x(k-1) + x'(k-1)T + \frac{1}{2}w_x(k-1)T^2 \\ x'(k) = x'(k-1) + w_x(k-1)T \\ y(k) = y(k-1) + y'(k-1)T + \frac{1}{2}w_y(k-1)T^2 \\ y'(k) = y'(k-1) + w_y(k-1)T \\ h(k) = h(k-1) + h'(k-1)T + \frac{1}{2}w_h(k-1)T^2 \\ h'(k) = h'(k-1) + w_h(k-1)T \end{cases} \tag{1}$$

where T is the time of node motion; ω_k is the perturbation noise to which the node is subjected at moment k, following the normal distribution with expectation of 0 and covariance of Q.

The Kalman filter state update equation is obtained from the above equation following [5]:

$$\begin{cases} \widehat{x}_k^- = A\widehat{x}(k-1) + Bu(k-1) \\ P(e_k^-) = AP(e(k-1)^-)A^T + NQN^T \end{cases} \tag{2}$$

where $\widehat{x}_k^-$ is the *a priori* estimate, which is the predicted value of the motion at the next update time; $\widehat{x}(k-1)$ is the maintained state at the previous moment; $u(k-1)$ is the systematic variable, usually assumed as 0 in this scenario; Q is the covariance of the node motion subjected to perturbation noise ω_k. The transfer matrix A is defined as:

$$A = \begin{bmatrix} 1 & T & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & T & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

The system noise matrix can be formulated as:

$$N = \begin{bmatrix} \frac{1}{2}T^2 & 0 & 0 \\ 0 & T & 0 \\ 0 & \frac{1}{2}T^2 & 0 \\ 0 & T & 0 \\ 0 & 0 & \frac{1}{2}T^2 \\ 0 & 0 & T \end{bmatrix} \tag{4}$$

The motion monitoring equation is then defined as:

$$\begin{cases} \widehat{x}_k = \widehat{x}_k^- + k_k(z_k - Hk_k(z_k - H\widehat{x}_k^-)) \\ k_k = \frac{P(e_k^-)H^T}{HP(e_k^-)H^T + FRF^T} \\ P(e_k) = (I - k_kH)P(e_k^-) \end{cases} \tag{5}$$

where z_k is the observed value of the position, k_k is the Kalman gain, $P(e_k)$ is the covariance matrix of the posterior estimation error e_k, R is the covariance of the difference between the observed and actual coordinates of the node, the transfer matrix is defined as:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

TABLE I
IMPROVED RREQ ROUTE REQUEST FRAME FORMAT

Message Type	Flag					Reserved Bits	Hops
	J	R	G	D	U		
RREQ ID							
Target node IP address							
Target node sequence number							
Source node IP address							
Source node prediction address							
Source node sequence number							

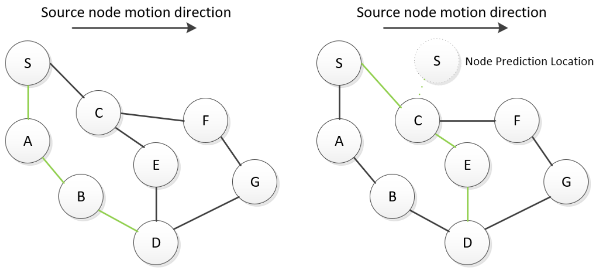


Fig. 1. Route discovery establishment process based on direction prediction

The observation noise matrix can be formulated as:

$$F = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \tag{7}$$

Now, the predicted coordinates of the node positions $(\widehat{x}_k^-, \widehat{y}_k^-, \widehat{h}_k^-)$ can finally be obtained from the predictions in the mutually orthogonal X,Y,H directions. The obtained predicted location is added to the Route Request(RREQ) frame. Before the source node transmits data to the target node, the source node needs to broadcast an RREQ message to the entire network in order to find the route address of the target node in the net [6]. Then, with this modification,the RREQ can effectively adapt to the node movement, reduce the process of repeated route discovery and increase the efficiency of routing.

The above Fig.1 shows a simple scenarios consisted with the source node S and the target node D along with other intermediate nodes to demonstrate the route discovery process. The left figure shows the route $S-A-B-D$ searched by the AODV route discovery process. Due to the rapid movement of the source node, the original routing method will change routes frequently. As a comparison in the right figure, the predicted position in the route request frame will replace the original node position. Then, the route will be re-planned with the path minimization strategy to replace the existing position with the predicted position. Now, the $S-C-E-D$ link route is established. It is obvious that, although the path is larger varied compared to the original route path, there will be a great extension in the life cycle of the formed route. Then, both the reduction in time delay and in the number of route discovery processes can be expected to save energy for intermediate nodes to survive longer in the network.

B. Improvement of routing maintenance algorithm based on dynamic repair packet and link quality monitoring

In order to ensure the link quality in the route discovery phase and the maintenance phase to find alternative routes for the next phase of route discovery, this section proposes an improved method in two main aspects: monitoring the link quality in the route discovery phase and finding alternative routes for existing routes rapidly. In the first phase, a log distance path loss model is introduced for monitoring the link quality in the route discovery phase [7].

$$P(s)(dB) = P(s_0) + 10nlg(s/s_0) + \delta \tag{8}$$

where $P(s)$ denotes the path loss in dB at a distance s between the transmitting and receiving nodes; $P(s_0)$ denotes the reference path loss at a close distance s_0 (often taken as 1 m) between the transmitting and receiving nodes, which is derived from the actual detection; n is the path loss coefficient decided by the surrounding environment; δ denotes a normal with standard deviation δ random variable, taking into account the environmental factors. Since the route discovery stage is to be preceded by node direction prediction before issue of the shortest path algorithm, s_0 in equation (1) will be the shortest path length. By letting $P(s_0)$ be 0, the path loss index $n = 1$, and the normal random variable δ can be avoided for simple scenario, the original equation can be simplified as follows [8]:

$$P(s)(dB) = 10lg(s/s_0) \tag{9}$$

In order to guarantee the link quality, the routing path length s should be limited to be $\sqrt{2}$ times the shortest path s_0. The path length s at this point is called the maximum selectable path s_max in the case of guaranteed link quality. When the improved routing path length is greater than s_max, the link quality of the path is considered to be poor. And an AODV route discovery policy maintenance is usually triggered to re-route discovery with the shortest path to re-route discovery.

$$P(s)(dB) = 10lg(s/s_0) > 10lg(\sqrt{2}s_0/s_0) = 1.51 \tag{10}$$

Now, when the path loss$P(s) > 1.51$, it indicates that the routing at this time is no longer guaranteed to meet the link transmission requirements. In this case, the route discovery interruption is triggered and the route will be re-selected using the path shortest algorithm.

In the second phase, the existing routes are quickly searched for alternative routes. The AODV route maintenance is improved by proposing a dynamic repair packet-based route maintenance strategy, which uses a repair packet forwarded with the data for real-time status monitoring and route maintenance of the link. When a route is established and data transmission starts, this repair packet is initialized by the source node and sent with the data. The Time To Live (TTL) of this repair packet is specified to be 1.

TABLE II
REPAIR PACKAGE FORMAT

Path loss P(s)	Path validity flag bit	Local node IP	Previous node IP	Next node IP	Target node IP	Hops

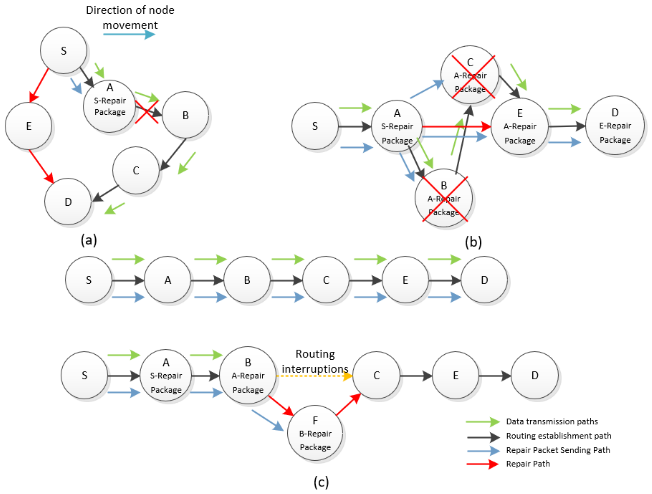


Fig. 2. Routing Repair

When the source node has data waiting for send, it generates a repair packet that is transmitted with the data transmission path. This packet will then be transmitted to the next hop along with the data [9]. Once the next hop node receives the packet, it modifies the address of the current node in the repair packet and generates a new repair packet named repair packet-A (i.e. suppose A is the name of the current node). If the node has more than one next hop nodes, it broadcasts the new repair packet to the next hop. If the node has only one next node, then the repair packet is transmitted with unicast to the next node. This process continues to generate new repair packets at the relied nodes until the destination node [10]. The proposed improved route maintenance operates in four main cases: interrupting invalid routes, adding routing nodes, deleting routing nodes, and routes that cannot be repaired, which will be discussed in detail:

1)Interrupt invalid routes

After receiving the repair packet, the node determines the path loss of the route and decides the value of the valid flag bit based on the size of the path loss, thus deciding whether the link is valid or not. In the case shown in Fig.2.(a), the source node establishes link $S-A-B-C-D$ by improving route discovery strategy. The source node starts to send data repair packet to target node D. And after receiving the repair packet, node A calculates the path loss $P(s)$ and finds that its value is greater than the maximum path effective loss. As a result, it takes the effective flag bit 0, which means the link quality is poor. Node A then feeds the flag bit to source node S , which triggers an AODV route maintenance and establishes a new link $S-E-D$ for data transmission.

2)Add routing nodes

In the interrupt routing, when a link is partially broken due to a shift in the network topology caused by the relative movement of nodes, it will be repaired by the contents of the repair package. As shown in the fig.2.(c), the established route $S-A-B-C-E-D$ is broken because of the relative movement of two of the nodes B and C, i.e. a break in the middle of the link, thus causing a break in the whole link. After receiving the repair packet from the front end, node B modifies the current node IP address information and broadcasts the newly generated repair packet outward. Then node F, which is in the next hop of node B, will receive the repair packet. After comparing the information embedded in the packet, it will found that the same next hop IP address exists in the next hop node of the point and in the repair packet. This indicates that the point can be utilized as an intermediate node between node B and C to repair the path. And the overall path loss will not change much after the path loss check, then node F is added as the repair node, and the data can be transmitted by the repair link path $S-A-B-F-C-E-D$.

3)Delete the routing node

The conditions for deleting routing nodes are exactly opposite to the scenarios of adding routing nodes. When the nodes moving towards each other in phase motion, the distance of the nodes becomes smaller. At this time, the routing can omit some unnecessary nodes, which can reduce the path loss. This will also reduce the number of hops, reduce the transmission delay, and ensure the efficiency of the link. As shown in the fig.2.(b), because of the phase motion, node A and nodes B, C, E can be directly connected, but the original link is $A-B-C-E$. If continue to follow the original route to transmit data, it will inevitably waste link resources. Therefore, when the repair packet is transmitted to A, A broadcasts the generated repair packet to its next hop node. Nodes B, C, and E will all receive the repair packet and all can reach the target node D. Therefore, comparing the hop count of several nodes to the target node in the repair packet, the one with the smaller hop count is used as the repair path, so the path becomes $S-A-E-D$ after the repair. And the redundant nodes are deleted to make the link transmission more efficient and reliable.

4)Routing cannot be repaired

If a route break cannot find a suitable relay node as a repair node to make the link complete the repair, the route will no longer be able to meet the demand and route discovery needs to be triggered to find a suitable route again.

III. EXPERIMENT RESULTS

A. Experimental simulation design

To demonstrate the efficiency of proposed method, we implemented a simulation program in Matlab. The main logic of the program is processed under the user interface. Once the user interface receives a request to send a packet, the program flows as follows: the data structure stores all messages sent over the network as part of the interaction. It contains 'node', 'nextNode', 'messageType', 'depth ', where 'depth' is the iterative value highlighted when displayed in the user interface (cyan links indicate the flooding process, blue links indicate the routing reply process, and green links indicate the data sending process) [11]. If the source node has a valid route to the destination, it tries to call the send function. If the sender encounters a routing error that causes the call to fail, the flood function is called. Both subroutines will write their changes to the hop count table. The flood call can be repeated if conditions warrant it. When the computation is complete, the hop count table is parsed and the routes are iteratively highlighted in the user interface to simulate data flow.

The send function is called starting from the source node until the current node is the destination node, it verifies if the current node has a route to the destination. And if the route is reachable, it repeats to the next node according to the routing table. In the case of failure, a routing error is sent backwards and the subroutine returns false. The flood function is then called to add the source node to the hop count table. The algorithm proceeds along the table reading and adding more entries. For each entry, the nodes that are not connected to the routing table are added to the table at depth + 1. On increment, the subroutine checks if multiple instances of the same node are added at the current depth. It removes all nodes except this is the node with the shortest hop distance. If a node is a destination or knows the path to the destination, it returns a routing answer before continuing. The subroutines terminate when traversing the entire table and return true only if an answer is sent at some point. the user interface is updated only after all subroutines have been called. The number of hops is displayed on the node graph, and the routing table saved by the node is updated. The size of each packet is to 1b for simple demonstration.

B. Experimental results and analysis

First set up a simple scenario of an aircraft communicating in an integrated network. The source node A starts a transmission over the network without any known conditions. An illustration of this scenario is provided in Fig.3(a).

Suppose the node A moves rapidly to the right side and sets up data to be sent from node A to the node F. Since node A has no route to F, the node first predicts the direction of motion, finds the next node located in the direction of motion. It then sends an RREQ packet out, and the request is forwarded by each node that receives it, thus propagating it throughout the network via flooding (represented by cyan links in the figure). In this process, each node sets up reverse routing entries detailing the next node, hop count and sequence number to be sent to the source of the routing request. After receiving the request, node F sends a RREP (represented by a blue link in the figure) to A in response. The intermediate nodes will know how to direct the route through the reverse route just established. Finally, if node A has a valid route to F, data and the dynamic repair packets will be sent

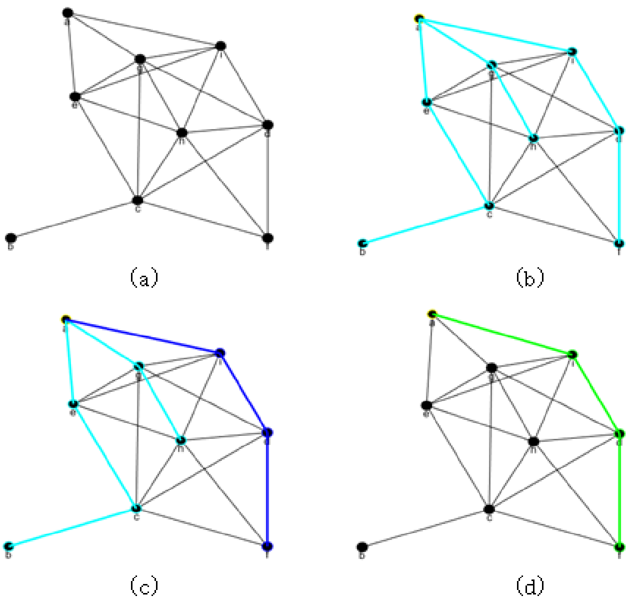


Fig. 3. Improved AODV protocol performance

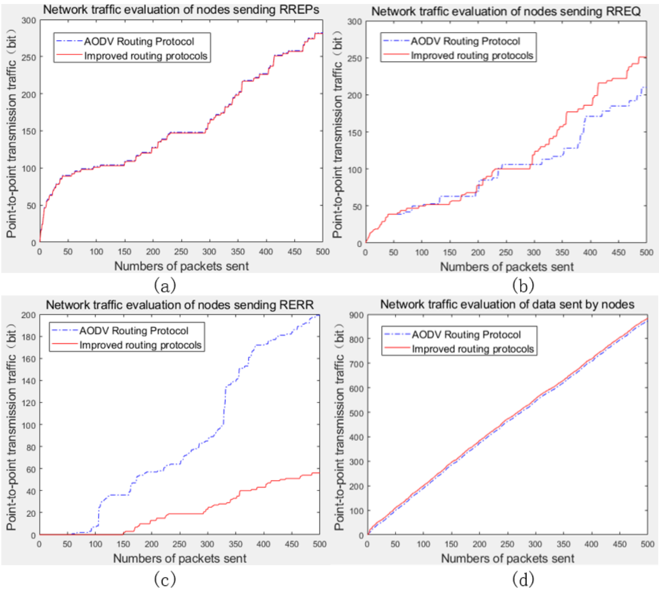


Fig. 4. Network traffic generated by nodes sending RREQ(Route Request), RREP(Route Reply), RERR(Route Error) and data

together (marked by a green link in the figure). Subsequent transmissions from A to F will continue to use this route with no new overhead until the link quality fails to meet the transmission requirements or the route cannot be repaired, and the route discovery process will be repeated.

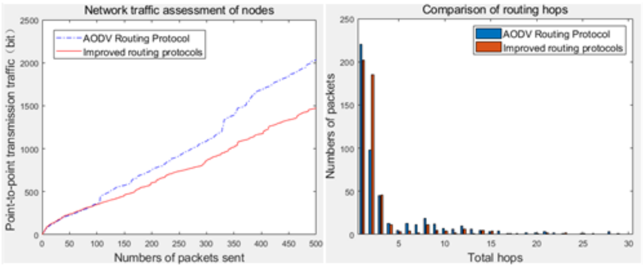
Fig.3.(b) then shows the route request process. From the figure, it can be seen that there are multiple routes to choose from A to F. For example, $A-E-C-F$, $A-G-H-F$ and $A-I-D-F$ are the three paths with the shortest number of entries. The node in the figure moves rapidly to the right. To accommodate the effect of this movement on routing, the path is found with the predicted node position. AODV routing protocol with shortest path algorithm path will be $A-G-H-F$, and the improved algorithm to accommodate the movement, the final path is determined as $A-I-D-F$. Then a reverse link is established for the target node F to A, as shown in Fig.3.(c). A valid route exists from the source node to the target node, and then data and dynamic repair packets will be sent for quality checking and route repair of the established link, as shown in the process in Fig.3.(d).

As shown in Fig.4.(a), the network traffic of RREQ messages generated by the two routing protocols sending 500 packets is compared. The number of RREQ messages steadily increases until approximately the 55th RREQ message, after which the difference between the two routing protocols begins to enlarge. As the packets continues to be sent, the network traffic for the route request messages continues to rise. At about 300 packets, improved routing generates slightly more traffic than the traditional routing. This is due to the traffic generated by the additional route requests in discovering the predicted direction of motion and selecting the next node. However, in the comparison of the total network traffic of the two routes, the improved route has a smaller network traffic, which we will mention in the following section.

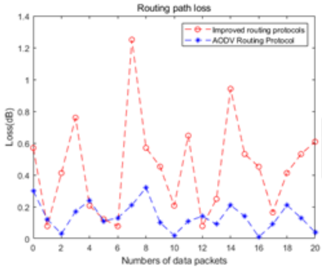
Fig.4(b) compares the network traffic of RREP messages generated by sending 500 packets by both routing protocols. Since RREP is a routing reply message sent from the reverse route based on the original route finding, the difference in network traffic performance between the two protocol schemes is not obvious, i.e. the network traffic of RREP messages generated by the two schemes sending packets are almost the same. However, in the RERR message, the difference in network traffic becomes very obvious, as shown in the network traffic diagram in Fig.4.(c). With the sending packets, the network traffic caused by RERR messages in the AODV route rises rapidly, while rises slowly in the improved route. This is due to the introduction of a dynamic repair packet for route repair in the improved route. The timely repair of the network can avoid the generation of too many RRER messages and ensures the stability of the link.

Fig.4.(d) shows the comparison of the network traffic generated by the two routing protocols when sending data packets. It can be noticed that the network traffic of the improved route is slightly larger than that of the AODV route when there is data transmission. This is because the improved route sends a dynamic repair packet at the same time as the data is sent, i.e. for monitoring the link quality as well as for route repair.

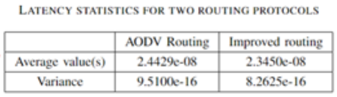
Fig.5.(a) the provide the total network traffic generated by the two routing protocols sending route requests, route answers, route errors, and data. The improved protocol has significantly lower traffic compared to the original protocol. For the improved routing, the node motion direction prediction is added in the route discovery phase, which can adapt well to the fast node motion. Although there will be more traffic in the route discovery phase and route repair phase, the additional



(a) (b)



(c)


LATENCY STATISTICS FOR TWO ROUTING PROTOCOLS

	AODV Routing	Improved routing
Average value(s)	2.4429e-08	2.3450e-08
Variance	9.5100e-16	8.2625e-16

(d)

Fig. 5. Total network traffic, hop count, path loss, latency statistics

cost of the direction prediction and repair packets is lower for the overall network traffic. This means the improved routing has better performance in adapting to the fast node motion.

To further compare the performance of the two routes, the hop count of routing data sent is compared. Again, nodes are set in motion and their positions are updated periodically, each with a predetermined step size and direction. The statistics were compared by sending the number of hops generated by 500 packets, and the results of the comparison are shown in the bar chart in Fig.5.(b). In the figure, both routing protocols hop counts are concentrated in the low hop count region. The traditional routing calculation method uses path minimization calculation to find routes, i.e. the routes established in this process are mainly concentrated in the part with low hop count, while the improved routing sacrifices some of the shortest paths for adapting to fast movement. Although the hop count is larger, it still distributed in the region with low hop count. Under the above simulation conditions, the same 500 data packets were sent for latency statistics comparison. The data characteristics are analyzed (Fig.5.(d)) and the expectation and variance of the two routing methods are derived from the data in the table. The results show that the improved routing has a smaller delay expectation and the delay data is more stable and less volatile. Fig.5.(c) shows the path loss comparison graph, sending 20 packets to compare the path loss between the improved routing protocol and the AODV routing protocol. The improved routing protocol has a greater path loss than the AODV routing protocol, but the loss is guaranteed to be kept below 1.51 dB in order to ensure link quality.

IV. CONCLUSION AND FUTURE WORKS

In this paper, the limitations of classical AODV algorithm in air-ground integrated networks have been analysed first, which reveals that the fast topology change of air-ground integrated network will cause frequent routing setup. As a result, not only the delay of data transmission will be enlarged, but also the maintenance packet exchanges will be increased as well. The Kalman filter aided AODV has been proposed consequently, which mode the movement of air nodes with linear approximation and obtain accurate predictions in short time scale. With such predictions, the classical reaction based AODV can be reformed into a proactive based algorithm, which will update the potential routing with the predication of Kalman filter in digital domain. As a result, the frequent routing maintenance can be avoided, which will lead to the increased efficiency of data delivery and the decreased maintenance cost. The additional involvement of the dynamic repair packets makes the link connection more stable, extends the link lifecycle, and reduces the latency caused by frequent route discovery. However, the channel model utilized to validate the proposed method only considers the path loss, which will be enhanced with more realistically air-ground channel model in our future works. The possibility to character the nonlinear movement will be focused as well in the future work.

REFERENCES

[1] A. Chistyakov, “A software architecture for large multi-simulation experiments over ad hoc networks using ns-3 discrete-event network simulator,” in *2014 European Modelling Symposium*. IEEE, 2014, pp. 403–408.

[2] I. Habib, N. Badruddin, and M. Drieberg, “A comparison of load-distributive path routing and shortest path routing in wireless ad hoc networks,” in *2014 5th International Conference on Intelligent and Advanced Systems (ICIAS)*. IEEE, 2014, pp. 1–5.

[3] C. G. Prevost, A. Desbiens, and E. Gagnon, “Extended kalman filter for state estimation and trajectory prediction of a moving object detected by an unmanned aerial vehicle,” in *2007 American control conference*. IEEE, 2007, pp. 1805–1810.

[4] H. Menouar, M. Lenardi, and F. Filali, “Movement prediction-based routing (mopr) concept for position-based routing in vehicular networks,” in *2007 IEEE 66th vehicular technology conference*. IEEE, 2007, pp. 2101–2105.

[5] G. Welch, G. Bishop *et al.*, “An introduction to the kalman filter,” 1995.

[6] C. Perkins, E. Belding-Royer, and S. Das, “Rfc3561: Ad hoc on-demand distance vector (aodv) routing,” 2003.

[7] O. Katircioğlu, H. Isel, O. Ceylan, F. Taraktas, and H. B. Yagci, “Comparing ray tracing, free space path loss and logarithmic distance path loss models in success of indoor localization with rssi,” in *2011 19thTelecommunications Forum (TELFOR) Proceedings of Papers*. IEEE, 2011, pp. 313–316.

[8] S. Ranvier, “Path loss models,” *Helsinki University of Technology*, 2004.

[9] I. D. Chakeres and E. M. Belding-Royer, “Aodv routing protocol implementation design,” in *24th International Conference on Distributed Computing Systems Workshops, 2004. Proceedings.* IEEE, 2004, pp. 698–703.

[10] J. Wu and I. Stojmenovic, “Ad hoc networks,” *Computer*, vol. 37, no. 2, pp. 29–31, 2004.

[11] A. Nakhaee, A. Harounabadi, and J. Mirabedini, “A novel communication model to improve aodv protocol routing reliability,” in *2011 5th International Conference on Application of Information and Communication Technologies (AICT)*. IEEE, 2011, pp. 1–7.