# Moth- and Ant-Inspired Routing in Hierarchical Airborne Networks with Multi-Beam Antennas

Lei Hu[1], Fei Hu[1]*, Sunil Kumar[2]
[1] Electrical and Computer Engineering, University of Alabama, Tuscaloosa, AL
[2] Electrical and Computer Engineering, San Diego State University, San Diego, CA
Email: {lhu7@crimson.ua.edu, *Corresponding author:fei@eng.ua.edu, skumar@mail.sdsu.edu}

***Abstract*—In this research we target the routing design in a directional hierarchical airborne network. Such a network has the following 2 main features: (1) *Two-level architecture*: The high level is a sparse wireless network with long-distance links and high-rate communications. It also has a commander node. The lower level is a high-density network with short-distance low-rate links. (2) *Hybrid directional antennas*: The network has mixed antennas (multi- or omni-directional antennas). We assume that the low-level nodes have inexpensive omni-directional antennas, while the high-level nodes are equipped with more expensive, multi-directional (also called multi-beam) antennas. We propose to use bio-inspired algorithms (based on moth and ant's behaviors), to achieve the high-throughput, low-latency two-level routing. Particularly we use male moth's light source pursuing pattern to handle the low-level network routing from an event node to the highly mobile 'sink' node, and use the ant's chemical trail maintenance principle to trace the trajectory of commander node in order to deliver low-level data to the high-level commander node (in high-level network) with the minimized delay. Additionally we propose to construct a weighted fence routing topology among high-level nodes with multi-beam antennas, to achieve a high-throughput routing in high-level network. Our simulation results show the significant improvement of the routing performance via bio-inspired routing, compared with conventional ad hoc routing schemes.**

***Index Terms*—Multi-Beam antennas, Hierarchical wireless network, Ad hoc routing, Bio-inspired Networking**

## I. Introduction

### A. Hierarchical hybrid wireless network ($H^2WN$)

We target the routing issues in a hierarchical hybrid wireless network ($H^2WN$). It has a typical two-level topology and hybrid antenna systems (omni- or multi-directional antennas). The high-level network has a sparse topology consisting of powerful nodes that can communicate with each other over long-distance links. Those nodes are equipped with multi-directional antennas to achieve multi-beam, high-rate transmissions. The low-level network has a much higher node density than the high-level network. Low-level nodes also have much lower mobility. To reduce the cost, they are equipped with omni-directional antennas.

Such a hierarchical network can be seen in many applications such as airborne network, smart factory, sensor and actuator network [1]. The reason of using such a hierarchical architecture is because of its separation of different types of wireless nodes, easy network management, and good balance between deployment cost and network throughput. For instance, we can use the expensive high-level nodes to deliver the high-throughput data that is generated from many low-level inexpensive nodes. Typically there is a 'sink' node in the low-level network that can collect event data from any node. The high-level network often has a node that serves as the role of 'commander' that can control the whole network.

The antenna technologies have been improved drastically in the past decade due to the rapid development of advanced circuits and materials. Fig.1 shows the typical antenna product appearances and their physical models.

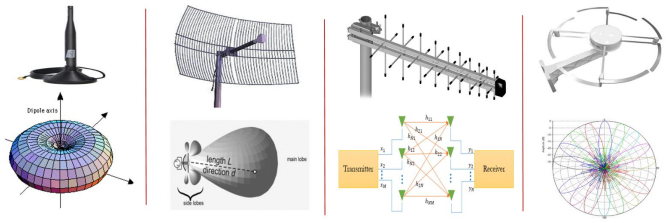


Fig. 1. Antenna models: (1) Omni-directional; (2) Uni-directional; (3) MIMO; (4) Multi-directional

The inexpensive popular antenna is omni-directional antenna (Fig.1 (1)). It simply radiates the same signal to all the directions (360 degrees) even when it only needs to communicate with a specific node. Thus it can cause strong RF interference among all the neighbors. To better focus the RF energy to a specific direction, we can use directional antenna with one beam, also called unidirectional antenna, see Fig.1 (2). Since all signal energy is concentrated on a small angle, it avoids the interference with other directions, and is also able to send data for a longer distance than omni-directional antennas.

Compared with MIMO antenna (Fig.1 (3)) which has complex antenna matrix coefficient control and needs the receiver's real-time feedback, another promising antenna, called *multi-beam smart antenna* (MBSA), see Fig.1 (4), has very simple antenna beam control. The MBSA simply extends the single-beam antenna to multi-beam architecture, and allows independent transceivers to be used in each beam. Each beam can thus send out different data based on its own queue management policy. Thus a MBSA significantly improves the node throughput through *simultaneous* data transmissions in multiple directions. In this paper we assume that the *high-level nodes are equipped with MBSAs*. Thus those powerful

nodes can achieve long-distance, high-rate transmissions. But the low-level network still uses omnidirectional antennas.

### B. Problem Statement of $H^2WN$ Routing Issues

This research targets the routing protocol design in the above described $H^2WN$. Particularly we will target 3 routing tasks: low-level routing, high-level routing, and cross-level routing (see Fig.2). Note that those 3 routing tasks can form a complete end-to-end path across the entire $H^2WN$: An unmanned aerial vehicle (UAV) can use 'low-level routing' to report its event data to a mobile sink that then uses 'cross-level routing' to reach a high-level aircraft, which can then use 'high-level routing' to reach another aircraft.

**Problem of low-level routing:** We will address the UAV-to-UAV routing issue in the high-density low-level network. Particularly, we assume that there is a UAV that serves as the *sink* node. If any UAV detects an abnormal event, it immediately forwards the data to the sink via multi-hop routing scheme. Although general event-to-sink routing has been studied [2], [3], they can not handle *the sink's singular mobility* problem. Unlike the high-level nodes that have 'even mobility', that is, each node has similar mobility (for aircraft network, it is generally $80 - 120m/s$), the low-level network has 'singular mobility', that is, most nodes may have little or low mobility, while the sink could have high mobility for the purpose of global data collection.

Note that the above 'singular mobility' issue is popular in many *mobile data collection* applications. The sink needs to move around to collect the data from the nearby nodes. If the sink is static, the nodes far away from the sink will take a long time to reach the sink. And it is also unfair to the nodes close to the sink since they need to help to relay the packets from all other nodes. Due to the singular mobility feature of the sink, all the nodes have similar probability of relaying the data.

To handle the sink's singular mobility, the conventional on-demand (*i.e., reactive*) routing such as AODV protocol could not fast track the sink's sudden leaving status, not mentioning the *proactive* routing (such as DSR scheme) that only works well for semi-static network topology. Since most of other low-level nodes do not have such sudden leaving behavior, the global path search via blind RREQ broadcasting could cause much routing overhead in a high-density, low-rate ad hoc network (such as our targeted UAV network here).

Our work will solve such an event-to-sink routing path search issue under singular mobility of the sink. Our routing scheme will be able to fast track the sink's movement trends. It first uses a multi-hop relay path to quickly reach the approximate 'sink area', and then uses gradient routing to deliver the data to the exact location of the sink.

**Problem of high-level routing:** We will also address the aircraft-to-aircraft routing issue in the high-level network. The biggest challenge is how to utilize the benefits of multi-beam antennas in the high-level routing process. Conventional ad hoc routing schemes such as DSR only use one path to deliver the data. Thus they need only one of the beams in the MBSAs for communications. General multi-path routing schemes [4], [5] often use widely disjoint paths to deliver data. Such disjoint paths do not have many intersection points among them. Since only those intersection points can possibly use multiple beams to deliver data, the nodes cannot fully explore all the available beams to improve the throughput. It is challenging to design a low-overhead, high-throughput, multi-beam routing among the long-distance high-level links.

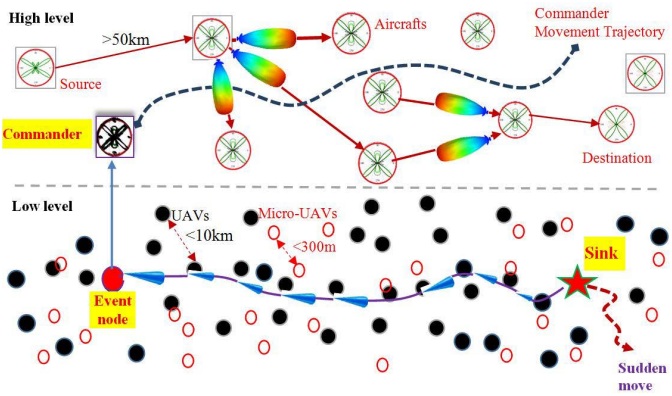


Fig. 2. Challenging routing issues in hierarchical directional wireless network

**Problem of cross-level routing:** Between the two levels, a critical routing issue is how an event node in the low level can quickly localize the *commander* node in the high level, and then efficiently forwards the data to a low-level node that is closest to the commander node. Note that the high-level network is much sparser than the low-level network. Thus it is difficult for an event node to find the closest high-level node. In order to deal with these issues, we develop a GPS-free commander node tracking scheme in order to efficiently find a low-level node that is closest to the commander node.

### C. Our Contributions

In this research, we will propose a series of innovative routing solutions to overcome the above challenging routing issues based on bio-inspired principles. Our contributions include 3 aspects as follows:

(1) *Fence-like high-level multi-beam routing*: We propose to build a fence-like routing scheme in the high-level network with a routing topology similar to the weighted neural network, to concurrently dispatch the data to multiple beams. Such a fence routing structure can explore the high-capacity of MBSAs very well.

(2) *Moth-inspired low-level routing*: Male moth has a peculiar trajectory when searching for a light source. It follows a straight line first, then follows zigzag curve, and finally uses circular trajectory to lock the source. Such a pattern helps to quickly localize an uncertain light source. Inspired by moth movement, we propose a line-fan-ring (LFR) routing search scheme to handle the sink's singular mobility issue in the low-level network.

(3) *Ant-inspired cross-level routing*: Ants use striking chemical scents to record the trajectory such that their partners can quickly find the food source. Such a trail has time-decaying feature. Inspired by ant chemical trail, we use high-density low-level nodes to record the trajectory of the commander

node in the high-level network, and create a time-decaying routing path to reach the commander node from any node in the low-level network.

**Roadmap**: The rest of this paper is organized as follows: In section II, we will briefly summarize other related works. Then section III gives our system assumptions. Section IV has the detailed discussions of *high-level* network routing scheme with a neuron grid architecture. We then move to the *low-level* routing based on moth-inspired algorithm in section V. The cross-level routing is described in section VI. The simulation results are provided in section VII, followed by the conclusions in Section VIII.

## II. Related Works

**On directional routing:** Most conventional ad hoc routing schemes assume that the networks are equipped with omni-directional antennas [6], [7]. In order to improve the throughput performance, directional routing has been proposed in a few works [8], [9]. However, most of them just consider the directional antenna with a single beam [10], [11]. Some reactive routing schemes such as AODV, search the routing paths only if these routes are required for a coming task. This helps to efficiently reduce the route maintenance overhead. However, for the networks with MBSAs, those conventional routing protocols have some dominant drawbacks such as traffic storm problem when searching for the new paths in the networks with high-density nodes. Those conventional routing schemes often assume the use of omnidirectional antennas, and do not have concurrent, well-scheduled data delivery features when MBSAs are used in the high-level network.

**On multi-directional routing**: In [12], the author proposed a multi-path data delivery scheme based on OFDMA and MIMO. However, MIMO has entirely different features from MBSAs (see Fig.1). We previously proposed a diamond-based routing protocol for the airborne network [13]. However, it cannot explore ALL the beams' capacities due to the use of a single relay node in some hops. In this work, we propose a fence-like routing protocol in which each hop has multiple nodes that can explore the multiple beams for concurrent transmissions.

**On bio-inspired routing:** In some of the recent works [14] [15] about hierarchical network routing, they assume a cluster-based network topology. But they do not consider the singular mobility of the destination, which may become a critical factor affecting the overall performance of the hierarchical network. In [16], a moth-inspired routing protocol is proposed to improve the communication performance in mobile networks. Under certain conditions this protocol may improve the throughput and reduce the packet loss rate in the network. However, it can easily select the non-optimal routing path, and the performance such as network delay and communication overhead may be sacrificed. In [17], a bio-inspired node localization mechanism for hierarchical networks is proposed. In that work, the current location of the mobile sink is projected into the 2D hull to maintain its location information. This mechanism can help to build a cross-layer routing path in the hierarchical networks. The geographic routing is used there and the performance of this protocol can be further improved by applying other mobile protocols such as DSR or OLSR. However, such a solution still could not handle the networks with MBSAs and singular mobility issue.

**On cross-level routing:** Currently there are some schemes that deal with cross-level routing, i.e., the routing issues when the source and destination are located in high and low levels, separately. There are mainly two types of $H^2WN$ routing designs so far:

(1) Routing in wireless sensor and actuator networks (WSANs) [18], [19]: WSAN is a typical $H^2WN$. Its low-level has a large amount of sensors, and high-level has a small amount of actuators. A low-level sensor may need to search a multi-hop path to reach a closest high-level actuator, in order to report its urgent monitoring data to the actuator to ask for its responses/actions. For example, in a digital farming system, when the humidity sensor detects a ultra-dry region, it asks the closest water sprinkler (i.e., an actuator) to perform irrigation. Routing design is WSANs is relatively easy since the most WSANs assume static sensors and actuators in both levels. Therefore, general routing schemes such as AODV or DSR can be used for cross-level routing in WSANs. However, such simple WSAN routing scheme cannot be used for our targeted $H^2WN$ here (such as an airborne network) due to the special network properties including highly mobile, MBSA-equipped high-level aircraft nodes and singular mobility of low-level sink node. In the simulation section (Section VII) we will show the poor performance of WSAN cross-level routing scheme when used in our airborne network scenarios.

(2) Routing in robot-based cyber-physical systems (CPS) [20]–[22]: Today many factories use robot-based CPS to achieve operation automation. Typically, the low-level network consists of a large number of sensors to detect the environment parameters, which will be transmitted to the high-level nodes - robots, for intelligent control of the manufacturing process. Unlike the above WSANs, the robot-based CPS allows certain mobility for the robots, although they are not as fast as the aircraft. Some mobility-adapted routing schemes are used for such a CPS. For example, OLSR can be used for cross-level sensor-to-robot communications. However, they typically do not use high-throughput multi-path routing in high-level robots. And there is no work on multi-beam-based routing design among robots. In section VII we will show that conventional robot-based CPS routing scheme cannot explore the high-capacity of MBSAs.

In a nut shell, no work has been conducted on the three challenging $H^2WN$ routing issues including the sink's singular mobility, multi-beam concurrent transmissions, and cross-level airborne network routing design.

## III. SYSTEM ASSUMPTIONS

This work targets a $H^2WN$ with hybrid antennas. Since omnidirectional antennas have been well studied before, here we explain the MBSA features as follows:

*Tx/Rx consistency*: All beams can send (Tx) or receive (Rx) data. And each beam could operate in different channels.

However, for those beams using the same channel, they should obey "beam consistency" principle, that is, those beams should be either ALL-Tx or ALL-Rx mode (Fig.3 Left). The reason of this constraint is due to the energy leaking from one beam's main lobe into another beam's side lobe if they are in different Tx/Rx modes. Of course, if they use different channels, there will be no energy leaking since there is no radio interference between different frequency bands (Fig.4 Right).

*Beam locking*: If a beam finishes its data transmission earlier than other beams, it can NOT change its Tx/Rx mode if there still exist other beams that are using the same channel to send/receive data. For this case, this beam can switch the channel in order to change its Rx/Tx mode.

Due to the above principles, it is critical to schedule the data transmission in the hop-to-hop routing path carefully among the high-level nodes, in order to fully explore the high-capacity of the MBSAs, meanwhile not violating the above communication constraints in different directions.

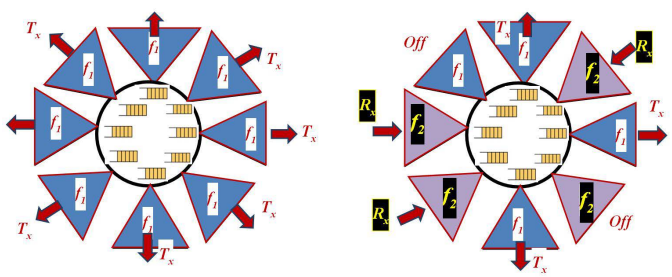


Fig. 3. Features of multi-channel MBSAs

**On the impacts of high mobility in the high-level network:** In airborne networks, the high-level nodes are aircrafts, which could have high mobility. However, such a high mobility does not mean that we cannot build a relatively stable routing topology. As shown in Fig.4, assume a beam has 60 degrees of coverage. The aircraft-to-aircraft distance could be $100km$ long. An aircraft with initial position in the center of the angle and the mobility of $120m/s$ will take average 8 minutes to fly out of a beam's scope ($58km$ long). For such a long time, any routing protocol could easily finish a typical communication session. Even though the light propagation delay is $0.3ms$ for $100km$ of link distance, a common DSR-based routing protocol just needs around $5ms$ of time to finish the propagation of RREP and receive the RREQ. Therefore, for airborne network, high mobility is not a main concern in terms of routing stability performance.

This work will use airborne network as a $H^2WN$ example. The proposed routing schemes can be easily extended to other $H^2WN$ applications. In the low-level airborne network, the UAVs have limited antenna gain and short radio propagation distance ($< 10km$). We assume that an UAV can only reach the aircraft **right above it**. If the commander node moves to another place, a UAV needs to search a low-level routing path to reach a UAV that is closest to the commander. This is the motivation of recording the trajectory of the commander by using the high-density feature of UAV network.

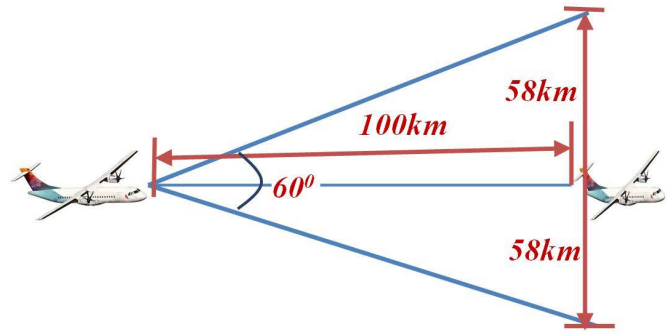


Fig. 4. Impact of high mobility

## IV. High-Level Network: Fence-Structured Multi-Beam Routing

### A. Problem Statement

The high-level network consists of powerful nodes with MBSAs. As an example, an aircraft can use high-gain MBSA to send data to multiple directions within a $> 50km$ of radius. The high-level routing should fully explore the benefits of multi-beam, concurrent Tx/Rx capabilities. However, none of existing ad hoc routing schemes explores the MBSAs' advantages.

As shown in Fig.5 (a), general DSR or AODV-like routing protocols only use a single-path to deliver data. They use at most 2 beams of the MBSA. One may argue that other source/destination pairs may help to utilize the multi-beam capability of a node. However, those pairs' paths may have very few intersection nodes (note that only those intersection nodes can explore their multi-beam transmissions). Handling multiple pairs in a single routing scheme (Fig.6(b)) would involve complex schedule coordination among those pairs, if we want to ensure that those intersection nodes follow multi-beam concurrent communication constraints. Furthermore, those pairs may have very different starting/ending times and QoS requirements. Thus it is difficult to coordinate the hop-to-hop relay schedules among all nodes, especially in those intersection nodes. In this work, we target the throughput maximization problem for a *particular* communication pair by using as many beams as possible to deliver data. Thus multiple pairs will automatically benefit from each individual, throughput-maximized communication pair.

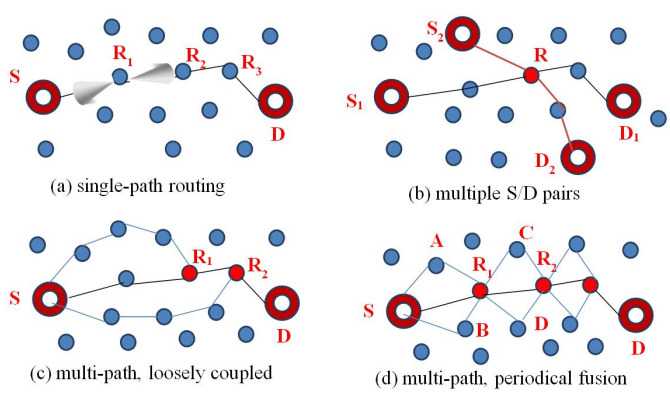


Fig. 5. Conventional routing schemes cannot well explore the MBSA's benefits

One may think of multi-path routing schemes that may possibly explore the MBSA capability. It is true that by using multiple, intersected paths we can at least explore the multi-beam capability for some of the nodes (such as nodes $R_1$ and $R_2$ in Fig.6(c)). However, those paths may be loosely coupled, and most relay nodes may still use at most 2 beams during their communications. A more coupled routing structure called Diamond Routing [13] is shown in Fig.6(d). It uses periodical traffic convergence/divergence nodes (such as $R_1$ and $R_2$) to guarantee that there always be a main path consisting of fusion nodes that can explore their multi-beam capabilities. Now the question is: how do we explore the multiple beams of the rest of nodes in such a diamond routing topology, such as the nodes A, B, C, D, etc., which may still use only 2 beams for Tx/Rx in diamond scheme? Our proposed fence routing will solve such an issue (discussed below).

### *B. Fence-Structured Routing Methodology*

Neuro scientists have attempted to understand how the human brain uses the largest intelligent network in this planet, i.e., neural network with billions of neurons, to memorize things. When we look at the findings of the neuroscience [23], [24], it is surprising to see that there exist essential commonalities between neural networks and MBSA-based routing:

*(1) MBSA-like tentacles help a neuron to quickly deliver pulses to neighboring neurons:* Using special cell microscopy, people have found that a neuron could have numerous tentacles in its ending location. Those tentacles look like a MBSA's beams. A neuron could fast pump different amplitudes of bio-electrical signals into hundreds of tentacles simultaneously. Those tentacles send or receive bio-pulses concurrently. This matches with MBSA's Tx/Rx consistency and beam synchronization principles.

*(2) The neurons use tentacles with different biological amplitudes ('weights') to form a weighted neural network:* A neuron's tentacles do not use the same pulse control level, that is, each tentacle has very different bio-pulse amplitudes. Using those 'weighted links', billions of neurons form a perfectly coupled neural network in the brain.

*(3) Human brain quickly recalls something by building a main neuron path:* Scientists have found that the brain is able to broadcast a bio-electrical query message to certain area and retrieves the results by enhancing a **main path** that consists of most relevant neurons. Such a main path is important since it helps to retrieve more and more detailed memory information from nearby neurons. It does this by extending the main path to a wider neuron path. This explains why the brain gives us an old event's big picture first, and then retrieves more and more details.

### *C. Concrete routing implementation: cross-level fence routing*

*1) Search for main path:* To determine the main direction and range of such a fence routing 'pipe', we will need to search for a *main path* first, just like the above main neuron-chain idea. Then later on we can add 1- or 2-hop nodes around the main path to form a grid-like fence routing pipe.

Although we can simply use DSR-like RREQ broadcasting to establish a main path, such a blind global broadcasting may cause packet flooding and high routing protocol overhead, especially for large-scale networks. Due to the cheap, easy-to-install compass, each node can easily determine $4$ directions (east, south, west, and north). If more than $4$ beams are available, a node can further distinguish more directions.

A simple routing scheme, called ORRP (orthogonal rendezvous routing protocol) [25], only uses two pairs of orthogonal directions (i.e., west-east and south-north) to search for a routing path. ORRP consists of two parts: (1) **proactive** maintenance of rendezvous node: a node maintains multiple rendezvous nodes which may be a few hops of away from the node. A node can easily appoint some nodes as rendezvous nodes by issuing R-RREQ messages to $4$ directions, and then periodically check the existence of those nodes. If they move too far away, they will be replaced with closer ones. (2) **reactive** establishment of an orthogonal path to one of the rendezvous nodes: If a source wants to search a path to its destination, it sends out RREQ in $4$ directions. If in one of its directions it reaches a rendezvous node, it will use that path as the routing path.

Since we only limit RREQ in $4$ directions, it is possible that the sender may never find a rendezvous node. However, in [26] it has proved that the failure probability is less than $4\%$. And such case only occurs when the network is too sparse. As a matter of fact, any two pairs of orthogonal lines have over $95\%$ of chance to intersect with each other [26]. If the network is too sparse and a rendezvous node cannot be found, conventional DSR-based blind RREQ broadcasting can be used to search for the *main path*.

**ORRP Enhancement via Two-time Launching Process**: The original ORRP could find a path that is much longer than DSR result. In this work we further improve ORRP by using two-time launching concept (Fig.6). In the first-time launching, we use the above described method to find two paths. Note that two pairs of orthogonal lines could have at least two intersection nodes. Thus we know that a shorter main path must be in such a rectangular area. Then the sender rotates about $45$ degrees toward the 'inside' direction (Fig.8), and performs the second-time launching, i.e., sending out RREQ message in the new direction until reaching a rendezvous node that is pre-maintained by the destination node.

Note that in a real MBSA it has at least $4$ beams (to cover $4$ directions). If the MBSA has $1$ more beam between any two directions (thus it has a total of $8$ beams), we do not need to rotate the sender's MBSA since the beam between the two directions could be used to launch the RREQ message. By using the above two-time launching, we could find a main path in the diagonal direction of the rectangular area. Such a main path is very close to the DSR result.

The search process for the *main path* based on the enhanced ORRP with 4-beam antennas is shown in *Algorithm* 1.

*2) Establishment of Fence Routing:* Based on the neural network principle, one can recover all the details of a past event by activating the nearby neuros around the main path. The links between the neurons have different weights, to reflect the fact that some life details could be remembered better than

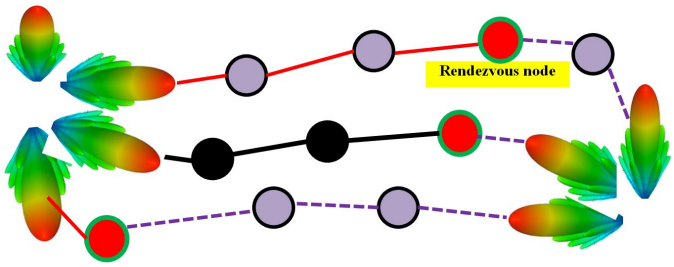


Fig. 6. Enhanced main path search via two-time launching

**Algorithm 1** Main path search based on enhanced ORRP scheme

*Input:* Source Node ID **S**, Destination Node ID **D**, two pairs of orthogonal Beam IDs

*Output:* The main path $P(\mathbf{S}, \mathbf{D})$ of the fence routing

**Part One:**

1) **D** periodically broadcasts ORRP announcement messages along the orthogonal directions from the 2 pairs of orthogonal beams.
2) Each neighboring node that received the messages stores the Destination ID **D**, the neighbor ID in the previous hop, the hop count and its Beam ID $BID_1$ from which the packets came in.
3) Each relay node forwards the announcement packets using the beam in the opposite direction.
4) Proactively, these relay nodes and destination node maintain the rendezvous-to-destination paths.

**Part Two:**

1) **S** sends out the route request packet (RREQ) in its 4 orthogonal beams.
2) The neighboring node that receives the RREQ packet stores the Source ID **S**, its previous hop, the hop count and beam ID ($BID_2$) from which it received this RREQ, then forwards it to the next hop using the beam in the opposite direction.
3) The subsequent nodes keep forwarding the RREQ along the orthogonal directions in the same way until a rendezvous node $R$ specified in Part One receives the RREQ. This intersection node $R_s$ stops forwarding and sends out a RREP containing the route to destination node **D** back to node **S** in the reverse direction.
4) After **S** receives the RREP packet, it establishes the source-rendezvous-destination path.

**Part Three:**

1) Based on Part Two, **D** rotates the beam facing the intersection node $R_s$ for about 45 degrees and then repeats the steps in Part One.
2) **S** also rotates the corresponding beam for about 45 degrees in the internal direction and repeats the steps in Part Two.
3) There exists a new rendezvous node $R'$, and **S** can achieve a shorter path which can be taken as the main path $P(\mathbf{S}, \mathbf{D})$ in the fence routing.

others. Those weighted neuros form a fence-like 'information pipe' to recover all the relevant memory signals.

We thus propose to build a fence routing based on the neural network principle. As shown in Fig.7, once a main path is found (here it is S-C-F-H-D), each node in the main path (except the source and destination) searches the 1-hop away neighbors. Note that here we ask the node to search the 1-hop neighbors in the direction that is orthogonal to its ORRP RREQ forwarding direction. For example, F uses its neighbor E as the fence node, C uses B and D, and H uses G. This is important in order to keep similar distances between neighboring fence nodes (such as B - E and E - G). Since the RSS (received signal strength) is proportional to the link distance in free space, the main path node always selects the node that gives itself a good RSS. It is possible that a main path node does not have any 1-hop neighbor in the orthogonal direction. Or, it may find only one neighbor in one side.

One may argue that a node could select 2-hop-away neighbors to serve as fence routing in order to establish more links among them. However, it has some drawbacks: (1) in some high-level network such as airborne network, the link distance could be over $100km$. If we select a longer distance node (such as 2-hop-away nodes), it may cause ultra-long links. This makes the radio propagation delay (at light speed) longer, and complicates the goal of synchronizing the multi-beam Tx/Rx transmissions due to the large link delay differences. (2) A node may only has 4 beams. If a beam needs to communicate with more than 2 nodes, it needs complex schedule control to avoid packet transmission collisions. (3) Too many nodes in the fence routing make the routing topology maintenance difficult. Therefore, here we only use 1-hop fence nodes.

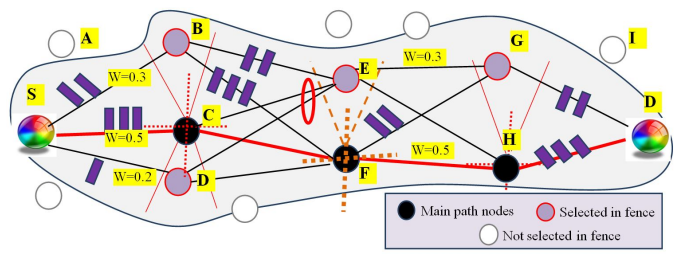


Fig. 7. Establishment of fence routing

Additionally, in the fence routing, each node should maintain a routing table as shown in Table I.

TABLE I
ROUTING TABLE

| Dest. | Beam | Next Hop | Link W | QoS | Applications |
|---|---|---|---|---|---|
| **F** | 2 | C | 0.50 | 1 | Video Conference |
| **B** | 1 | B | 0.30 | 3 | Live Stream |
| **C** | 2 | C | 0.50 | 2 | Video on Demand |
| **D** | 3 | D | 0.20 | 4 | File Download |

Note: every node keeps local sense of directionality. QoS can be introduced in the routing protocol.

**Link weight determination based on cross-layer design:** since the link weight will be used to allocate the traffic amount in each link (more traffic will be assigned to the beam with better link conditions), we will need to define the link weight accurately. Here we use 3 parameters (one from each layer) to define the weight:

$$W = W_1 * RSS + W_2 * (1/PER) + W_3 * (1/ETX), \quad (1)$$

where

$$W_1 + W_2 + W_3 = 1. \quad (2)$$

Here RSS is a Physical layer parameter, and $RSS$ is proportional to $1/(d^2)$. packet error rate (PER) is a Data Link layer parameter:

$$PER_j^{(k)} = 1/(1 + e^{\eta * (SINR_j^{(k)} - \sigma)}). \quad (3)$$

Here, we denote the packet error rate (PER) of channel $k$ for a node with priority $j$ as $PER_j^{(k)}$. $\eta$ and $\sigma$ are constants corresponding to the modulation and coding schemes for a given packet length. ETX (expected transmission count) is a

Router layer parameter, and has been defined in the literature such as [27]. The above weight definition reflects the cross-layer design nature of our routing design.

*3) Cross-layer MAC/Routing design:* To fully explore the multi-beam concurrent transmissions of MBSAs, the above fence routing scheme should be integrated with the MAC layer schedule control. To fully explore the benefits of MBSA-based routing, we recommend that the MAC layer use a TDMA-like schedule control between different hops. If the MAC is CSMA-based scheme, a coarse TDMA-based sublayer could be added above the CSMA. Since the TDMA-like schedule uses much longer slot duration (could handle over 10 packets) than general TDMA time slots, adding such a higher sublayer is not a big issue. The cross-layer design is important for a few reasons:

a) Achieve a pipelined transmission: As shown in Fig.8, suppose Col-1 nodes are receiving data from S, Col-2 nodes can send data to Col-3, and Col-4 can send to D. The nodes belonging to the same column will have similar transmission delays after we allocate the traffic to each link based on the link weights. For example, if S-to-2 link's weight is double of S-to-3 link, we can allocate double traffic amount in S-to-2. Because we use rateless codes that can automatically adjust the sending rate, the S-to-2 link would have approximately double sending rate than S-to-3, thus the transmission delay is approximately the same in those two links. Therefore, we could make all nodes in the same column achieve a coarse Rx/Tx synchronization. If Col-(i) is receiving data, it cannot send data to Col-(i+1). However, Col-(i+1) can send data to Col-(i+2) at the same time. Thus a pipelined column-by-column transmission is achieved.

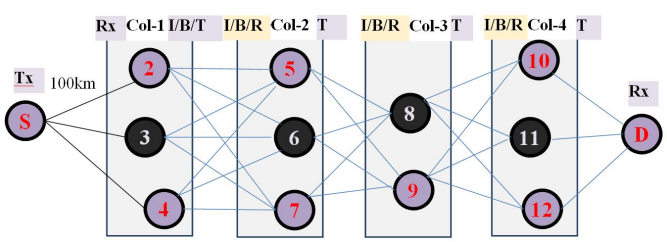


Fig. 8. Cross-layer MAC/Routing Schedule Control

b) Handle long-distance links: The above mentioned TDMA-like MAC layer is important to some $H^2WN$ applications such as airborne networks. Due to the long-distance link, the propagation delay could not be ignored. For example, if a link is $100km$, it is $0.33ms$ of delay. For such a 'long' time, it is difficult to accurately detect transmission collisions if CSMA-based MAC scheme is used. To avoid the waste of link bandwidth, some packets could be sent during such a long time. Therefore, some researchers [28] suggest to use TDMA-like scheme to send some packets each time in a Tx phase, and then switch to Rx phase to receive some packets. Since the TDMA slot time is much longer than conventional slot definitions, it cannot be called real TDMA scheme (instead, it is called TDMA-like scheme). Additionally, the time boundaries between Tx/Rx do not need to be accurately determined. In other words, there may exist certain Tx/Rx starting time deviations among different nodes in the same column. We call such a situation as *coarse synchronization*.

The high-level network may not have such a long link. For example, in a small-size wireless sensor and actuator network, the distance between the actuators may be only $500m$. For this case, in MAC layer, CSMA-based scheme can still be used. A higher MAC sublayer can be added above the CSMA to achieve a coarse TDMA-like Tx/Rx switching control. For example, in [28] it has proposed the overlay network concept to use long time interval in the higher sublayer.

c) Explore the multi-beam feature for reverse data transmission: In Fig.8, when Col-1 nodes are receiving data from S by using left half of its beams, the right half of its beams could be in idle (I) status; Or, they could receive data sent in reverse direction (either ACK packets or reverse data sent from the receiver). We use 'B' (back) to mean reverse direction data.

*4) Bandwidth Allocation Policy:* The bandwidth allocation policy is defined as a linear programming problem with certain network and interference constraints [29]. We model the fence topology as a directed graph $G(V, E)$. For simplicity, we assume the largest width in the fence routing topology is 3, which means that in each column there exist a main path node and 2 side path nodes (one in each side of the main path). Thus for each node $v \in V$, there are at most three beams in active Tx or Rx mode. We denote a beam $i$ of node $v$ as $v_i \in v$. For any link $(u, v) \in E$, the flow rate or average throughput is denoted as $f(u, v)$, and $f(u_i, :)$ denotes the outgoing flow rate from beam $i$ of node $u$; $f(:, v_j)$ denotes the incoming flow rate to beam $j$ of node $v$. If node $v$ is within the range of beam $i$ in node $u$, i.e. $v \in R(u_i)$, the flow rate $f(u, v)$ can also be represented as $f(u_i, v)$. Assume there is a lower bound $f_0$ for the flow rate at each beam of the nodes in the fence routing topology. Additionally, we need to consider the radio interference. Our objective for bandwidth allocation is to maximize the overall throughput while minimizing the interference in the network with multi-beam antennas. We can now formulate the LP problem as follows:

$$max \sum_{S_i \in S} f(S_i, :) - IF \tag{4}$$

Subject to

$$\sum_{S_i \in S} f(S_i, :) = \sum_{D_j \in D} f(:, D_j). \tag{5}$$

$$f(u_i, :) = f(u, v) \quad for \quad v \in R(u_i). \tag{6}$$

$$f(:, v_j) = f(u, v) \quad for \quad u \in R(v_j). \tag{7}$$

$$\sum_{A_i \in A} f(A_i, :) = \sum_{A_j \in A} f(:, A_j) \quad for \quad A \in M. \tag{8}$$

$$f_0 <= f(u, v) <= C_e \quad for \quad (u, v) \in E. \tag{9}$$

$$Size(B) <= 3 \quad for \quad B \in V. \tag{10}$$

$$f(u, v) >= f(u, w) \quad for \quad (u, v) \in ME, (u, w) \notin ME. \tag{11}$$

Here, $C_e$ denotes the link capacity, $M$ represents the nodes in the middle of the fence except for source $S$ and destination $D$. And $ME$ is a link of the main path, which is supposed to have more bandwidth than the links of the side path. $Size(B)$ represents the number of beams in use for node $B$ when

$B$ is transmitting or receiving data. $IF$ represents the total interference in the network.

*5) Overall high-level network routing protocol:* We provide the main operations of the routing protocol among the high-level nodes as follows:

**INPUT:** Node IDs, beam IDs, Channel IDs, Traffic QoS parameters (delay deadlines), link rate and network capacity in each beam, source traffic amount and rate, antenna beam angle, node mobility speed;

**OUTPUT:** Fence routing topoloy, Tx/Rx schedule

**Protocol Part 1 - Main/side paths establishment**

1. Using the enhanced ORRP algorithm described before to establish a GPS-free main path.

2. Every main path node $h_i$ in stage $i$ widens the path to construct side paths by searching neighbors in both sides of the main path via neighbor discovery scheme. Also a upper bound $V_{max}$ and lower bound $f_{min}$ are used to rule out the neighbor nodes with low flow rate and high mobility.

3. A side node in current *column* (also called *stage*), $i$, can be included in the neighbor list of the main path node $h_i$, as long as it is in one-hop range of the main path node. Suppose the list size is $N$. For the desired stage width $W$ (it means that there could be the maximum W nodes in one stage), if $N > (W-1)$, we could select $(W-1)$ side nodes with relatively higher sending rate and lower mobility among all $N$ nodes. If $N < (W-1)$, then the width for this stage needs to be reduced. For example, assume that the desired stage width is 4, however in stage $i$, there are only 2 side nodes in the list. Then in stage $i$ the width has to be reduced to 3. Therefore there may be different widths for every stage.

4. Building all the stages until reaching the destination node $D$. Every node in stage $(i+1)$ should be within the communication range of all the nodes in stage $i$. This is also the reason that we cannot use a too high value of $W$. Otherwise, a node in stage $i$ may not be able to read a stage $(i+1)$ node in the diagonal direction.

**Protocol Part 2 - Beam orientation adjustment**

5. Based on the steps above, the stages are constructed to form a fence structure. Then for a MBSA of any node, the beam direction should be adjusted properly to establish the links between the nodes in two consecutive stages. A node $A$ in stage $i$ keeps sending probing messages through its beams to the nodes in stage $(i+1)$ , if a node $B$ in stage $(i+1)$ receives the probing message, it sends back an ACK message. Thus the link between $A$ and $B$ is established, and the beam of $A$ stays towards $B$.

**Protocol Part 3 - Beam table maintenance**

6. After the beams' directions are determined in the above step, the beam table will record the node ID within each beam's range. In addition, in the probing ACK message, each node of stage $(i+1)$ feedbacks its information such as node ID, reception beam ID, signal strength, queue size, etc.

7. Each node in stage $i$ also puts its own information into the beam table including the beam ID, desired destination ID, beam range, queue length, traffic QoS, traffic type, traffic priority, etc.

**Protocol Part 4 - Traffic Control**

10. For any node, all the beams should have coarse time synchronization due to the concurrent packet transmission (CPT) and concurrent packet reception (CPR) requirement for all the beams of a MBSA. Between the stages the transmission (Tx) /reception (Rx) modes alternate, i.e. if stage $i$ is in Tx mode, then stage $(i+1)$ should be in Rx mode, and stage $(i+2)$ should be in Tx again.

11. To be compatible with TDMA-like protocol in MAC layer, the time interval for each Tx/Rx mode should be determined based on the traffic amount in each beam and the link speed. The goal is to ensure that the entire stage has coarse time synchronization.

**Protocol Part 5 - QoS Control**

12. For each node , the links in its beams are assigned different priorities based on the link quality (measured by *weights*). The links with higher priority are assigned with more traffic amount.

13. To avoid traffic congestion, the nodes in stage $(i+1)$ periodically inform the nodes in stage $i$ about the queue size.

## V. Moth-Inspired Low-level Network Routing

### A. Problem Statement

While the high-level network has even mobility (i.e. the nodes have similar moving speeds), the low-level network has **singular mobility**, i.e., the sink node typically has much higher mobility than other nodes. This is mainly due to its global network data collection requirement.

Let's first consider the case that the sink does not have much mobility. A simple *gradient routing* could well support the node-to-sink communications. As shown in Fig.9, the sink can simply broadcast an announcement message to the global network in order to know how many hops away each node is. Since the sink does not move much, it only broadcasts such a message occasionally. Then any node can simply forward data to an inner circle neighbor, and eventually reaches the sink.

However, if the sink moves much more quickly than other nodes, such a gradient-based routing does not work well since an established gradient map quickly becomes invalid when the sink moves to a new place within a short time.

Conventional DSR-like routing can not efficiently handle *singular mobility*. Since DSR uses blind, global RREQ broadcasting, a source node needs to frequently broadcast such RREQ messages in order to keep track of the fast moving sink.

Here we assume a GPS-free network. Thus the geographical routing based schemes will not work here either.

### B. Moth's Target Approaching Behavior

Moth has a special way to pursue the light source. It first detects the approximate direction of the light source (it has an acute thermal detection capability to find such a direction). Then it uses 3 steps to locate the light (Fig.10): (1) *Fast, straight approaching*: It first uses a diving speed to quickly approach to the target. (2) *Zigzag trajectory*: When the moth feels that the target is not far away, it slows down its speed and carefully flies toward the target with the zigzag style. (3) *Circular fly to lock the target*: When the moth feels that the

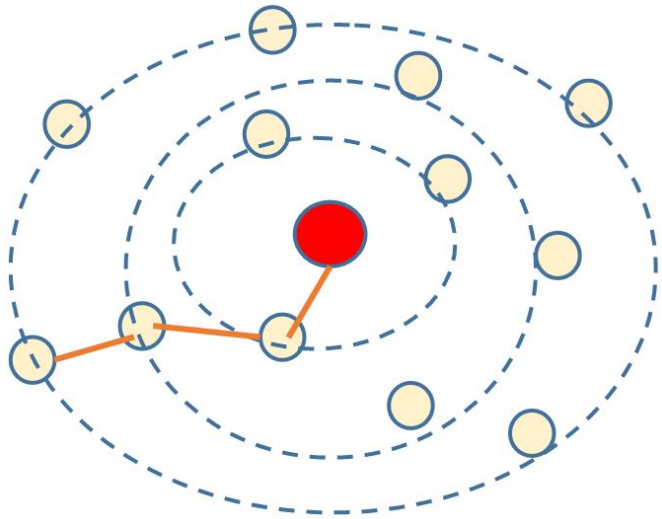

Fig. 9. Gradient routing

light source is close enough, it uses a circular trajectory to get closer and closer to the target until finally it locks the exact location [16]. Scientists think that the moth's innate capability is an efficient way to locate a target. Its behavior is similar to the spaceship's flying style: it first uses the highest speed to fast approach to a planet. Then it slows down and flies circularly in the orbit of the planet until landing.

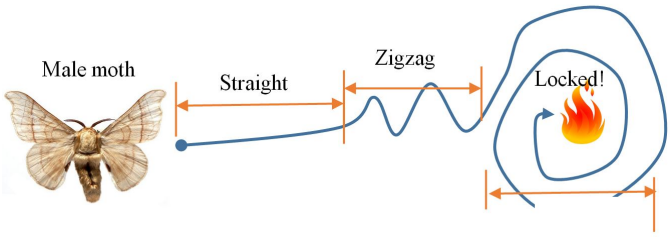


Fig. 10. moth's innate capability for an uncertain target

### C. Moth-Inspired Event-to-Sink Data Delivery

Inspired by the above moth's target searching behavior, we propose a line-fan-circular (LFC) routing path establishment scheme (Fig.11). The entire routing path consists of 3 parts: (1) *Line segment (LS)*: In this section of path, the event node (E) uses a single path to relay the data in hop-by-hop. (2) *Fan segment (FS)*: in this phase, the last node of the line section uses DSR-like RREQ broadcasting to search for the neighborhood of the sink. (3) *Circular area (CA)*: This is the neighborhood of the sink. When a FS node intersects with any of the CA nodes, it stops FS, and the intersection node uses gradient path to directly reach the sink.

Due to singular mobility, the sink has a quickly changing neighborhood. It is not beneficial to maintain a gradient routing map for the entire network since a node may never reach the sink if following an outdated gradient path. Therefore, we only ask the sink to maintain a very small gradient map (maybe just 2 hops away), see the circular area of Fig.13. Suppose a sink moves at speed of 100 meters per second. Assume

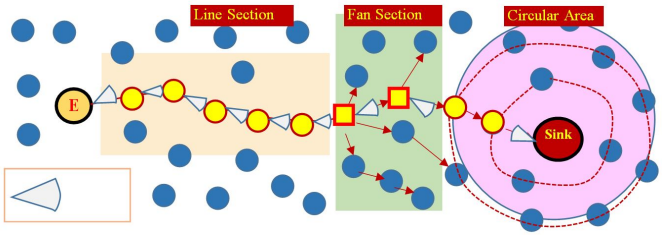


Fig. 11. LFC routing path establish scheme

that it broadcasts a message to its neighborhood every 10 seconds. Since the UAV communication distance can be up to 10km, the sink only leaves its original location for 1km within 10 seconds. The CA established last time is thus still valid. Therefore, as long as a packet reaches one of the CA nodes, it can be delivered to the sink.

The radius of a CA (measured as the number of hops) depends on the empirical values of the network settings. If the sink moves faster, it should maintain a smaller CA. Otherwise, it may expand its CA to more hops.

Whenever the sink wants to establish a new CA, it will broadcast a $CA\_Fresh$ message to its neighborhood. There is a field in the header called $CA\_TTL$ (time-to-live). Suppose it is set to 3. Each time a node receives the $CA\_Fresh$, it subtracts 1 from $CA\_TTL$. If it is zero, the node knows it is in the outer circle of the CA and stops further broadcasting.

Any node can at least use a *part* of the DSR-like path to reach the sink. Here we assume that the sink broadcasts the RREQ to the global network at a reasonable rate. After a complete DSR process through the network, all nodes will get to know the approximate gradients to the sink. Especially they will know the shortest path to reach the sink (i.e. along the gradients). Even if the sink quickly moves away, at least the first part of the path will bring the data to the correct direction to the sink.

This fact motivates us to use LS (Fig.13) to fast approach to the CA. The length of the LS depends on the empirical values. Its value is a trade-off between the following factors: the network scale, how many hops between the source and sink, the sink mobility, and the maximum 1-hop communication distance.

The FS is needed to reach the CA. In the moth's trajectory, there is a zigzag section between LS and CA. Here we make a light change: we use a short section of fan-like message broadcasting in FS section to search the CA. As long as the message reaches a node of the CA, the node immediately sends back a $CA\_FOUND$ message to the source.

Such a FS is very short compared to the LS. If LS is updated due to a new round of global DSR process launched by the sink, the corresponding FS path should be re-built. If the intersection node (with the CA) is not available anymore due to the sink's new CA announcement, the FS should also be updated to find a new intersection node.

### D. Moth-Inspired Routing Algorithm

Here we provide the big picture of the moth-inspired routing algorithm.

**INPUT:** node IDs, average node mobility $V_s$, sink mobility $V_d$ , sink announcement round ID $\#N_L$, sink announcement period $T_d$, link capacity $L_C$ and rate $L_R$, source traffic rate $R_S$, radio reaching range R , LS hop number $\#M_1$.

**OUTPUT:** Moth-inspired routing topology

**Periodical sink existence announcement:** The sink periodically broadcasts its information such as the current location and node ID to the network, by using DSR protocol (or other reactive ad hoc routing schemes). The nodes receiving the sink's message update their neighbor lists and routing tables to build the gradient routing paths. Those nodes use gradient routes to connect with the sink in the CA.

**Route partition with mobility:** The whole routing path is partitioned into three sections (LS, FS, and CA). In LS, the hop number $\#M_1$ depends on node mobility rates $V_s$, $V_d$. The nodes within the first $M_1$ hops starting from source node are selected from the DSR-based path to form the LS. The end node $A$ in LS is also regarded as the starting node of the FS. Node $A$ uses blind message broadcasting to searches for the target nodes $B_1, B_2, \cdots, B_n$ located in CA. One of the target nodes $B_i$ is chosen as the intersection node between FS and CA, if the path from $A$ to $B_i$ is the shortest among all the paths originated from $A$. By using gradient routing, node $B_i$ can reach the sink with singular mobility.

**Routing path update:** Every $T_1$ seconds, the sink broadcasts a $LocationUpdate$ message to nearby nodes (within $M$ hops), to build a new CA. If the target node $B_i$ is not in the new CA anymore, a new search of FS needs to begin to a find a new target node $B_j$. Every $T_2$ seconds ($T_2 > T_1$), the source triggers a new round of DSR to establish a new LS.

## VI. Ant-Inspired Cross-Level Routing

### A. Problem Statement

Although the sink could serve as the data aggregation point to collect event data from any low-level node, in many $H^2WNs$, one of the high-level nodes (called commander node), plays a more critical role than the sink since it can send commands to any low-level node (using downlink communications). It also collects emergency reports from any low-level node (using uplink communications).

Although downlink communication (commander-to-UAV) is relatively easier due to the powerful long-distance communication capability of the aircrafts, the uplink communication (UAV-to-commander) needs to adopt the principle of "the closer, the better" to save the transmitter's power. This is because an UAV is designed to communicate within a short-distance link ($< 10km$). It needs to use a high antenna power to reach a high-level node. Therefore, if it needs to talk with the commander, it must first forward the data to a UAV that is closest to the commander, and then asks that UAV to relay the data to the commander.

Figure 12 shows such an scenario. Suppose a low-level node $A$ wants to send data to a commander. Assume in the beginning it can send signals to the commander that is right above it ($20km$ away). Then the commander moves away for $100km$. Now A needs to propagate the signal for a much longer distance (this example is $102km$). Obviously, this is not realistic to a UAV with limited antenna power. A better approach is to use a node that is closest to the commander (here it is node $B$), to help to relay the data to the commander.

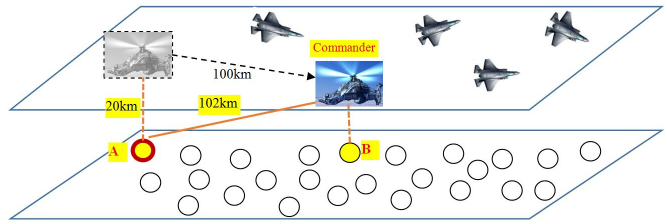


Fig. 12. cross-level communication

Therefore, always finding a closest node plays a critical role in UAV-to-commander uplink communications. Although some existing cross-level protocols are introduced to deal with the coordination issues between the high level and low level nodes (see Section II), however these cross-level schemes are not designed for multi-beam networks and they cannot address the singular mobility problem. Although some of them take the node mobility into consideration, they cannot fast track the highly mobile sink node. But our moth-inspired protocol well solves this issue. Our above described cross-level routing scheme can ensure the shortest uplink communication distance by recording the commander's latest locations through the low-level high-density nodes.

### B. Ant-Inspired Cross-Level Routing

Ants have the innate capability to collaborate to find food. As shown in Fig.13, when an ant finds a food source, it gets back to the nest and leaves a special chemical material called pheromone. Other ants will follow such a trail to reach the food source. When more and more ants follow the same trail, each of them leaves certain amount of pheromone trail, and thus such a trail is enhanced. If the food is already stored in the nest, the pheromone evaporates and eventually disappears. Then no ant will follow such a trail any more.

Inspired by the pheromone trail phenomenon [17], we propose to use the low-level nodes to record the trail of the commander node of the high-level network. Our idea is motivated by the following facts: (1) the low-level network is much denser than the high-level network. So many swarming UAVs can well record the trajectory of an aircraft. (2) The high-level node can easily reach the low-level node through its powerful antenna (a MBSA). Each time the commander moves, it can broadcast a TRAIL message to the low-level network. Any node receiving such a message will keep a record in its commander tracking table. Figure 14 illustrates the basic idea. Each trail node maintains a table containing beam direction, decaying factor, trail timestamp, next-node, etc.

**Ant-inspired cross-level routing protocol:** Here we provide the big picture of ant-inspired UAV-to-commander routing scheme.

**Input:** node IDs, link rate $R_U$ on the upper layer, link rate on the lower layer $R_L$, antenna signal range $R$, the

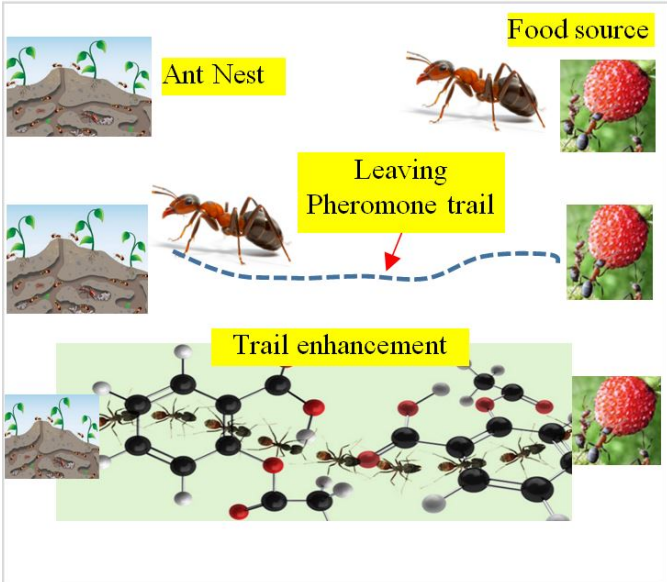


Fig. 13. pheromone trail phenomenon

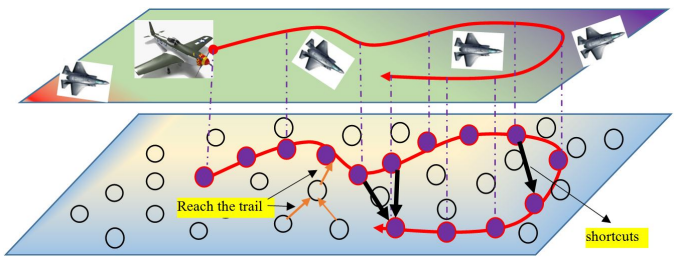


Fig. 14. ant-inspired cross-level routing

shortest distance between the low- and high- level network $h$, commander announcement period $T_P$;

**Output**: trail list in the lower level, cross-level routing path.

*Step 1*: The commander node $N_c$ periodically sends the $TrailNotification$ message with its ID and timestamp to the nodes in the lower layer. If a node $L_i$ in the lower layer receives the message from the commander node, then it creates a trail table to store the information including timestamp, last trail node ID, etc. Also node $L_i$ periodically broadcasts the trail table to its neighbors in the lower layer. Any node that hears the broadcasted trail table and is in the trail will add a "shortcut pointer" to its trail table to indicate that it can directly reach the ending node in the trail.

*Step 2*: If a node in the trail wants to send data to the commander node, it will first search "shortcut pointer". If the pointer exists, it will immediately forward data to the ending node in the trail. Otherwise, it initiates a route discovery to find a closest trail node. It does this by following general DSR process: it broadcasts a RREQ message to nearby nodes. If a node that received the RREQ and is in the trail, it feedbacks the $FoundTrail$ message ro the source.

*Step 3*: Decaying of the trail: if a node finds itself has not received the new $TrailNotification$ message for a duration that is longer than a preset threshold $T_{trail}$, it sends out a $LeavingTrail$ message to nearby trail nodes, and leaves the trail.

## VII. Performance Evaluations

### A. Fence Routing with MBSA

*1) The case of airborne network:* We first simulate a two-level airborne network, and the higher level is a sparse aircraft network with at most 5 hops between any source and destination. All the high-level nodes are randomly located in a area of $300km \times 300km$. The average link distance is $50km$, and the radio transmission range is less than $90km$ for a MBSA. The average transmission speed is 10 Mbps for the main path links, and 5 Mbps for the side path links. Every node has a buffer that can hold at most 200 packets. Each MBSA has 4 beams. We compare the throughput and delay of fence routing with diamond routing [13] and single path routing (DSR).

The throughput and end-to-end delay are shown in Fig.15 and Fig.16. The throughput of fence routing is higher than diamond routing and single path routing. The throughput of fence routing can almost reach the average packet generating rate, which indicates a negligible packet loss rate. The throughput becomes steady at around the generation rate of $1000packets/s$. But for diamond routing and single path routing, the steady throughput is only about $400packet/s$ and $200packet/s$, respectively. Regarding the end-to-end delay, fence routing has much less delay than other two schemes.

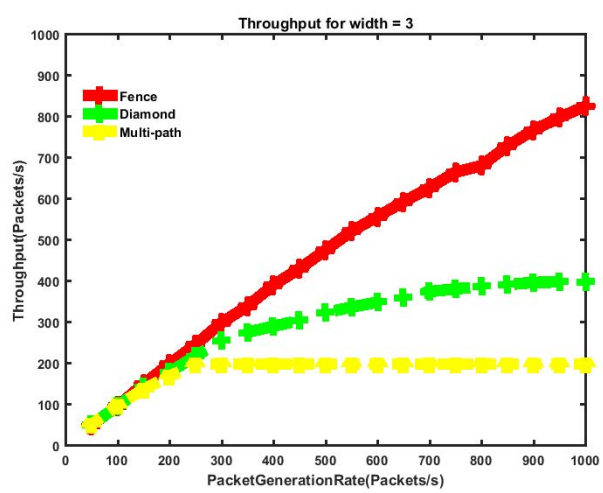


Fig. 15. Throughput (width =3)

We then study fence routing with the maximum $width = 2$ in each column. This means that in each column/stage, besides the main path node, only one side node can be added to the fence topology. Fig.17 and Fig.18 show the throughput and delay. Again, fence routing can improve the network performance compared with single path routing and diamond routing.

*2) General two-level wireless network:* Here we simulate a general network that could have over 10 hops of nodes between the source and destination in the high-level network with MBSAs. The link state is 1Mbps for the main path. Also the average link distance is $1km$ and the antenna transmission range is about $1.8km$ in each beam.

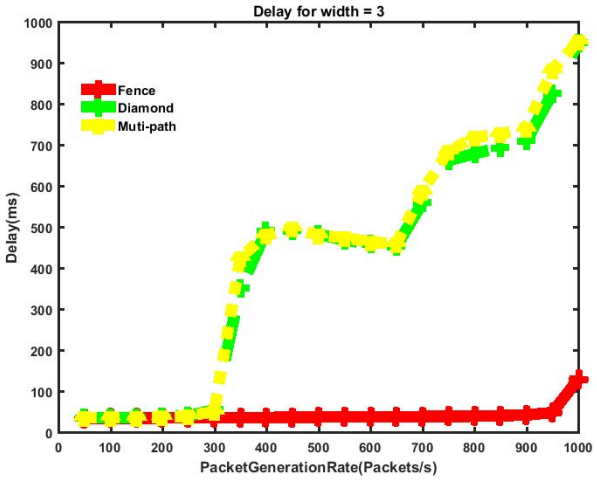


Fig. 16. Average delay (width =3)

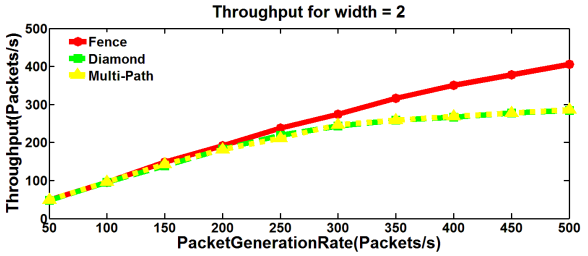


Fig. 17. Throughput (width = 2)

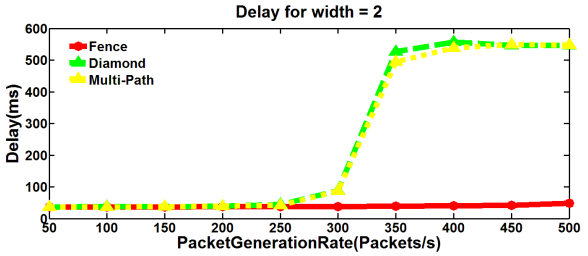


Fig. 18. Average Delay (width = 2)

Fig.19 and Fig.20 show the throughput and delay with width = 3 (i.e. there are 3 nodes in each column of the fence routing). Again, fence routing is much better than other two schemes.

We further investigate the impact of the maximum fence width on the performance for fence routing. Fig.21 and Fig.22 present the throughput and delay for $width = 2$ case. We can see that the throughput is lower for $width = 2$ than $width = 3$ case. This is because for $width = 3$ there are more nodes involved in the multi-beam forwarding. Larger width also reduces the delay.

### *B. Moth Routing in the lower level*

In the lower level of the hierarchical network, we will compare the performance of three routing protocols, including conventional static routing (DSR without path updating scheme), dynamic routing without considering singular mobility of the sink node, and our proposed moth routing). For the conventional mobile routing scheme, it searches for the new path again every 40 seconds. In the moth routing protocol,

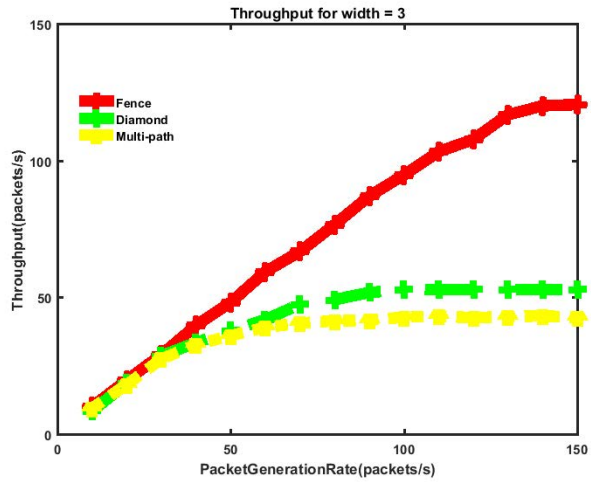


Fig. 19. Throughput (width = 3)

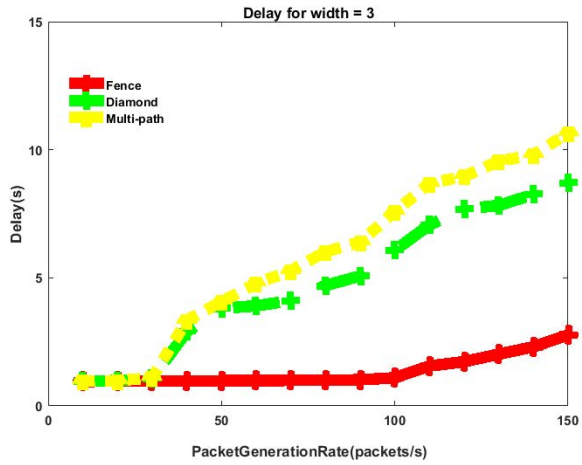


Fig. 20. Average delay (width = 3)

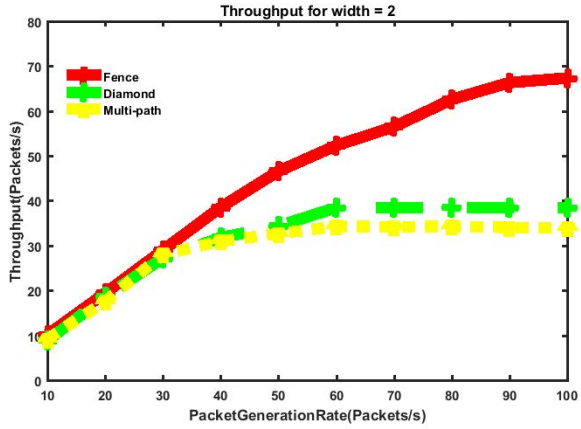


Fig. 21. Throughput (width = 2)

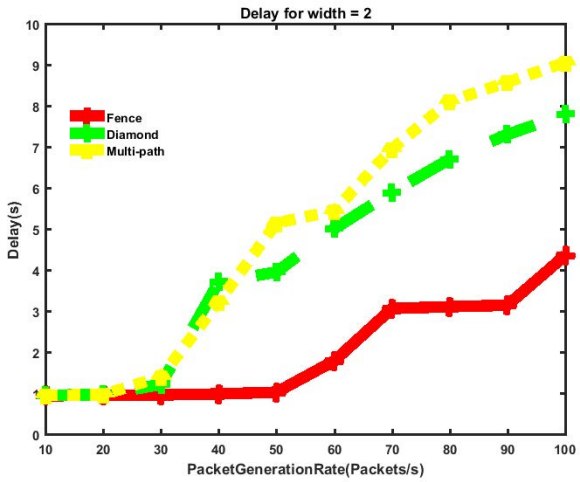


Fig. 22. Average delay (width = 2)

the sink broadcasts its message to its 2-hop neighbors every 10 seconds. And the packet generating rate at the source is $25 packets/s$. Assume there are 5 hops in the LS of moth routing.

Fig.23 shows the end-to-end throughput for different destination/sink speeds. At a low speed such as $0.1km/s$, the conventional mobile routing seems to achieve better performance than the other two protocols. This is because that moth routing may construct a non-optimal path compared with conventional DSR-like protocol. However, with the increase of the sink speed, the throughput of conventional mobile routing decreases significantly, while our proposed moth routing can still keep the throughput above $18 packets/s$.

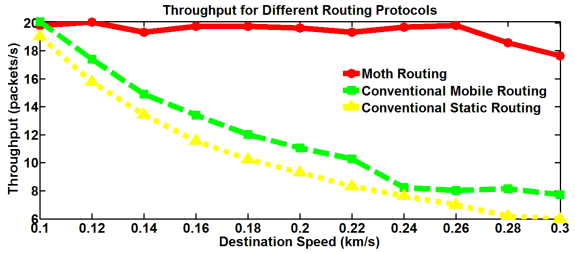


Fig. 23. Throughput of different routing protocols

Next we study the impact of the route updating period on the performance. Assume a sink speed of $0.3km/s$. From Fig.24 we can see a smaller updating period brings a higher throughput. We can see that $10s$ is the best route updating period for moth routing with a sink speed of $0.3m/s$, and the throughput can reach $15 packets/s$. Fig.25 shows that the throughput of conventional routing is normally below $6 packet/s$. For the same sink speed, moth routing can achieve higher throughput than conventional mobile routing.

**Different Update Periods for Moth Routing:** As we can see from Fig.26, in low sink mobility such as $0.1km/s$, updating period has small impact on the performance of moth routing. When the sink speed increases, updating period becomes more influential. From Fig.26 and Fig.27, we can see that 10s is a good choice of updating period for moth routing and achieves a throughput of 18 packets/s.

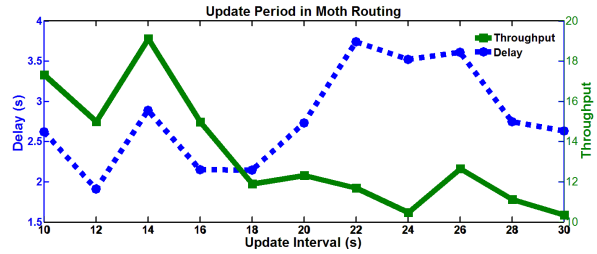


Fig. 24. Performance of Moth Routing

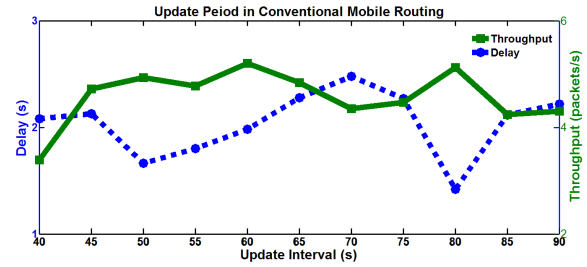


Fig. 25. Performance of Conventional Routing

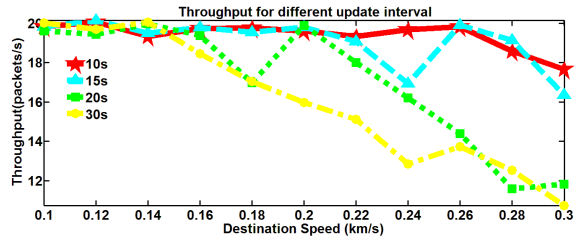


Fig. 26. Throughput for different updating periods in Moth Routing

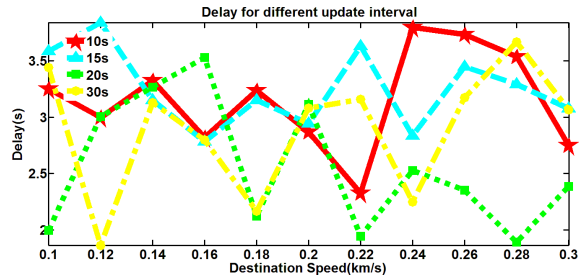


Fig. 27. Delay for different updating periods in Moth Routing

**The impacts of different LS lengths:** In moth routing, the first section is a straight line (LS) which helps to quickly approach to the sink's direction. From Fig.28 we can see that there exists a threshold for the number of hops in LS. In this particular example, if more than 6 hops are used in LS, the throughput will decrease significantly, also the delay will increase sharply.

**The impacts of different CA lengths:** To study the proper size of the CA in moth routing, we evaluate the performance of the overall network for different hops in this area. Fig.29 shows the average throughput for 3-hop CA, 2-hop, and 1-hop, respectively. We can see that the CA with 2 hops can obtain the best throughput. Thus in our routing protocols, the 2-hop neighbors are maintained in the CA for the gradient routing.

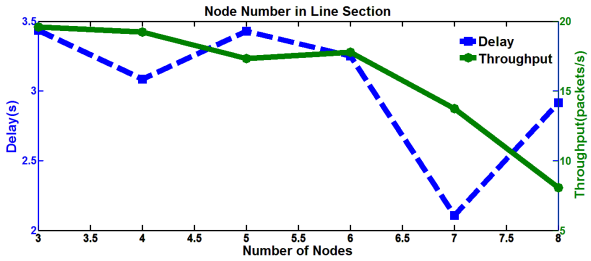


Fig. 28. Different Hops in Line Section

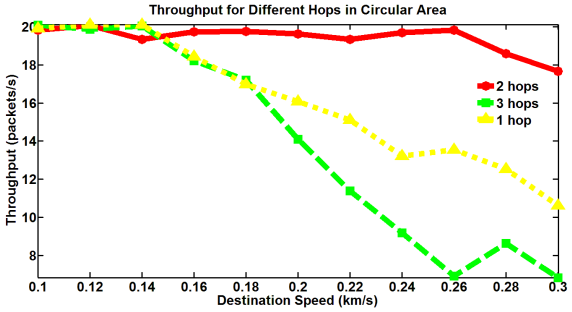


Fig. 29. Different Hops in Circular Area

### C. Ant-inspired Cross-Level Routing

In this section we will validate the ant-inspired cross-level routing. We assume that there is a commander node $C$ in the upper level and a source node $S$ in the lower level. And originally the source node $S$ is right below the commander node $C$. This means that node $C$ is in the direct communication range of node $S$, as shown in Fig.30(a). But after some time, node $C$ may fly out of the range, as shown in Fig.30(b). We have proposed an ant-inspired, cross-level routing scheme for the communications between $S$ and $C$. In the simulations, we consider a movement area of $20km \times 20km$ for both levels of networks. There are 10 nodes in the upper level and 100 nodes in the lower level. We assume that at first the commander node flies in the direction that is 45 degrees deviating from X axis. Then the commander adjusts its direction randomly with at most $\pi/16$ deviation from its original direction. We keep a trail list for the commander node by resorting to the nodes on the lower level. Fig.31 shows the number of hops in the routing path. We can see that as the commander node's speed increases, the hop number also increases (assume no shortcut scheme is applied). The hop number reaches 15 when the commander speed is $0.3km/s$. By using shortcut path, we can see that the hop number can be decreased to only 3 or 4 hops. Therefore with shortcut scheme, the complexity of our cross-level routing can be significantly decreased. We can also see that the commander's speed has little impact on the hop number with shortcut scheme.

Fig.32 shows the probability that node $S$ can directly reach the commander node $C$ without using the cross-level routing protocol. When the commander's speed increases, the probability decreases dramatically from 0.25 to 0.05. Since the probability for cross-level communication is lower than 0.25 most times, there is a high chance that the node $S$ cannot directly connect to the commander $C$. Thus our proposed cross-level routing protocol is important.

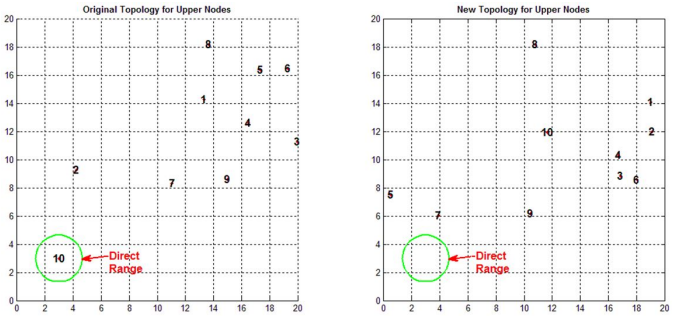


Fig. 30. (a)Original Location for Upper Nodes (b)New Location for Upper Nodes

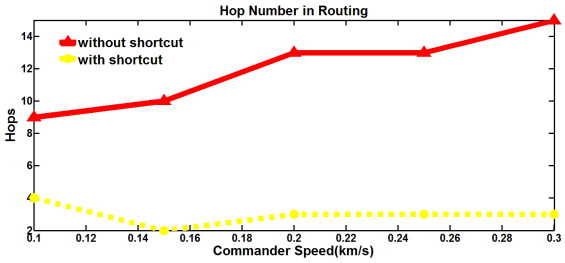


Fig. 31. Hop Number in the Routing Path

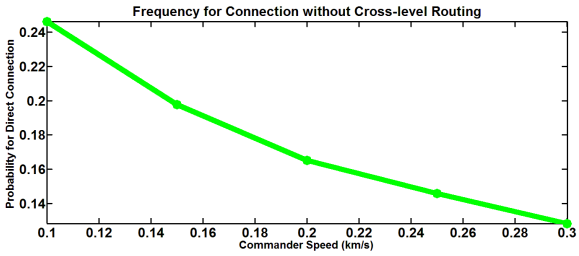


Fig. 32. Probability of direct connections without cross-level routing protocol

**Compare with other cross-level routing schemes:** As discussed in Section II, there are already some cross-level routing protocols besides our proposed bio-inspired cross-level routing. In our simulations, we use the same upper level and lower level network topology as used in the previous simulations. Then we implement the conventional cross-level routing schemes used for (1) wireless sensor and actuator networks (WSANs) and (2) robot-based CPS. Fig.33 shows the packet loss rates for 3 different protocols (i.e., our moth-inspired routing, WSAN sensor routing and CPS robot routing). We can see that our proposed moth routing can achieve a packet loss rate of below 0.2 no matter how high the sink's mobility speed is. However for the other two cross-level routing schemes, the packet loss rate can be pretty high when the sink mobility increases. Fig.34 shows the event-to-commander delay in the airborne network under those 3 cross-level routing protocols. Here we assume that one of the high-level nodes (could be an actuator in WSAN or a robot in CPS) serves as a commander. We can see that the delay for our cross-level routing scheme is much lower than the delay for the other two schemes. This indicates that our proposed bio-inspired cross-level scheme can ensure the high QoS performance and better exploits the

benefits of the multi-beam transmissions for the hierarchical airborne networks.

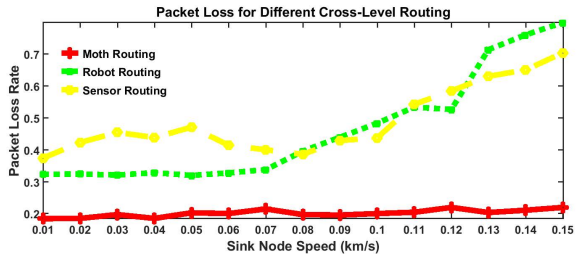


Fig. 33. Packet Loss Rate for Different Cross-Level Routing Protocols

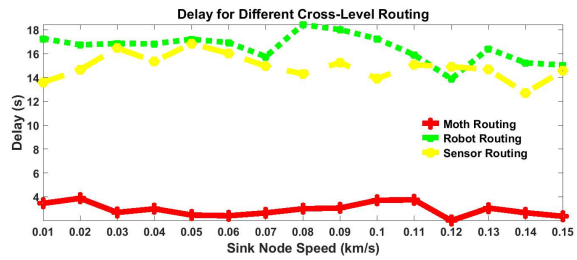


Fig. 34. Delay for Different Cross-Level Routing Protocols

### *D. Overall Network Performance*

In order to evaluate the end-to-end routing performance from a low-level source node to a high-level command node, we have integrated all the three proposed routing algorithms together, which include (1) the moth-inspired low-level UAV-to-UAV routing, (2) ant-inspired UAV-to-aircraft (low-to-high level communications), and (3) neuro-inspired fence routing for high-level aircraft-to-aircraft routing. And we have put significant effort to conduct new simulations for the end-to-end throughput and delay in the hierarchical network. Also, we compare our end-to-end routing algorithms with the existing hierarchical routing protocols such as mobile routing protocols and static routing protocols. Because the high-level aircraft network has much longer link distance (over 50km) than low-level UAV networks (less than 10km), we assumed 3 hops of the fence routing in the high level with the link speed of 10 Mbps. Because the link speed in the low level UAV network is just 0.5 Mbps, the throughput of the low level network becomes the bottleneck for the overall network.

Fig.35 and Fig.36 show the simulation results for three different protocols for the hierarchical networks (our proposed scheme and other two schemes): (1) Our Routing: This is our proposed end-to-end routing scheme that consists of three algorithms in the overall network including moth routing in the lower level, ant-inspired cross-level routing with shortcut, and fence routing in the higher level. (2) Mobile routing: This is a scheme that we will compare with. In this scheme, we apply mobile routing protocols in the whole network including conventional AODV-based mobile routing in the low level, multi-path routing in the high level, as well as our proposed cross-level routing. (3) Static routing: This is another scheme that we will compare with. In this scheme, we apply static routing protocols in the network which consists of static routing in the low level, multi-path routing in the high level, and our proposed cross-level routing.

In Figs.35 and 36, on the X-axis, the gateway node means the low-level node that can directly connect to the closest node on the high-level network. This gateway node helps to send the data from the low-level UAV network and the corresponding commander node in the aircraft network. Thus the mobility of the gateway node is critical since it determines the ant-inspired trail dynamics in the low-level network. We can see that our proposed routing has higher throughput and lower delay than the conventional protocols, especially when the gateway node is in high mobility speed. Our scheme can keep the throughput more than 10 packets/s even when the speed of the gateway node reaches as high as 0.3 km/s, however the throughput for the conventional protocols are significantly compromised in such a mobility speed.

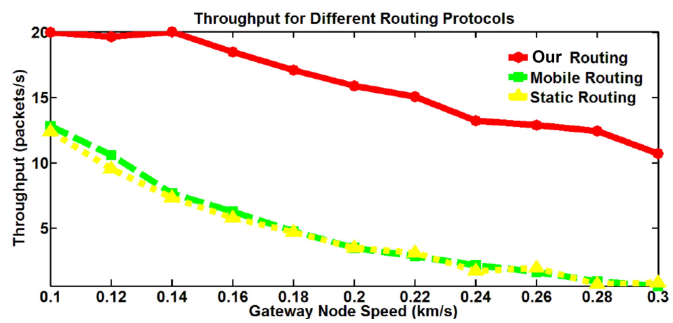


Fig. 35. Throughput for the overall network

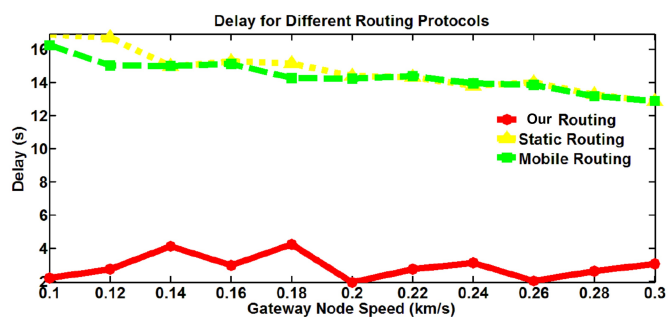


Fig. 36. Delay for the overall network

**Scalability of the routing scheme:** In order to study how scalable the proposed algorithms are, we have conducted new simulations for the hierarchical networks with different size to see if our proposed protocol can still take effect in large-scale networks.

We first increase the number of the aircraft nodes (from 25 to 50) in the high-level network, and enlarge the area from 300km x 300 km to 500km x 500km. Then we conduct the simulations on the new network topology for our proposed fence routing, diamond routing and conventional multi-path routing. The figures below show the performance of the new network topology. We can see from Fig.37 and Fig.38 that for the new large-scale network, our proposed fence routing can still get better throughput/delay performance than the conventional diamond and multi-path routing schemes.

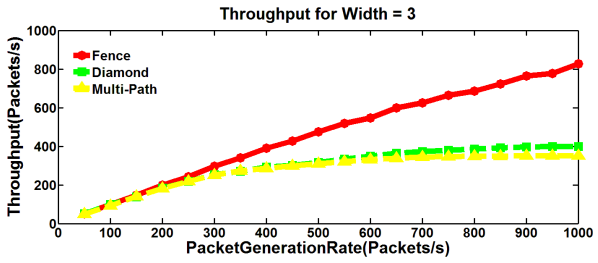


Fig. 37. Throughput for New Topology

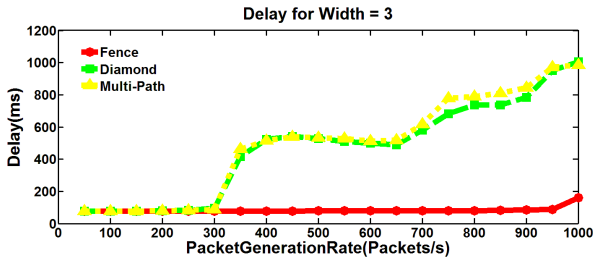


Fig. 38. Delay for New Topology

Then we increase the number of nodes of low-level network from 100 to 225 and also enlarge the size of the network topology. Again we conduct the simulations for this new topology for our proposed moth routing, conventional mobile routing and static routing respectively. Fig.39 shows the simulation results for this scaled network topology. We found that our proposed moth routing can still achieve better performance than the other two protocols in large-scale UAV network.

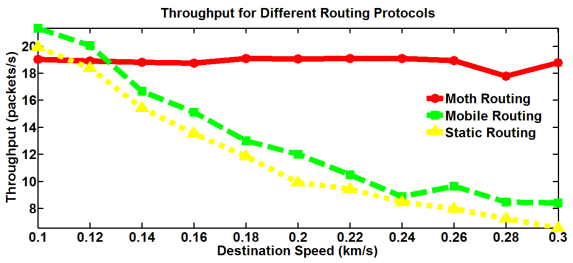


Fig. 39. Throughput for Different Protocols for New Topology

## VIII. Conclusions

In this paper, we have presented three novel routing protocols for the hierarchical network with multi-beam directional antennas. For the high-level network, we proposed a fence routing protocol that could efficiently exploit the benefits of multi-beam directional antennas. Our results showed that fence routing could achieve higher throughput and lower delay than other protocols. For the low-level network, we have designed a moth-inspired routing protocol composed of three different sections, i.e. line section, fan section and circular area, to overcome the sink's singular mobility. The simulation results showed that moth routing can achieve higher throughput and lower packet loss rate than conventional mobile routing protocols. We then used ant-inspired scheme to implement a cross-level routing protocol. Our results showed that the shortcut-enhanced trail-leaving scheme could significantly increase the reaching probability to the commander node.

## References


[1] M. Kawato, Y. Uno, M. Isobe, and R. Suzuki, "Hierarchical neural network model for voluntary movement with application to robotics," *IEEE Control Systems Magazine*, vol. 8, no. 2, pp. 8–15, 1988.

[2] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly-resilient, energy-efficient multipath routing in wireless sensor networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 4, pp. 11–25, 2001.

[3] B. Krishnamachari, D. Estrin, and S. Wicker, "Modelling data-centric routing in wireless sensor networks," in *IEEE infocom*, vol. 2, 2002, pp. 39–44.

[4] Y. Ganjali and A. Keshavarzian, "Load balancing in ad hoc networks: single-path routing vs. multi-path routing," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 2. IEEE, 2004, pp. 1120–1125.

[5] I. Cidon, R. Rom, and Y. Shavitt, "Analysis of multi-path routing," *IEEE/ACM Transactions on Networking (TON)*, vol. 7, no. 6, pp. 885–896, 1999.

[6] A. B. Malany, V. S. Dhulipala, and R. Chandrasekaran, "Throughput and delay comparison of manet routing protocols," *Int. J. Open Problems Compt. Math*, vol. 2, no. 3, pp. 461–468, 2009.

[7] S. Lu, L. Li, K.-Y. Lam, and L. Jia, "Saodv: a manet routing protocol that can withstand black hole attack," in *Computational Intelligence and Security, 2009. CIS'09. International Conference on*, vol. 2. IEEE, 2009, pp. 421–425.

[8] T. Song, W. Xia, T. Song, and L. Shen, "A cluster-based directional routing protocol in vanet," in *Communication Technology (ICCT), 2010 12th IEEE International Conference on*. IEEE, 2010, pp. 1172–1175.

[9] Y. Zeng, K. Xiang, D. Li, and A. V. Vasilakos, "Directional routing and scheduling for green vehicular delay tolerant networks," *Wireless networks*, vol. 19, no. 2, pp. 161–173, 2013.

[10] S. Bandyopadhyay, K. Hasuike, S. Horisawa, and S. Tawara, "An adaptive mac and idrectional routing protocol for ad hoc wireless network using espar antenna," in *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*. ACM, 2001, pp. 243–246.

[11] B. Hu and H. Gharavi, "Directional routing protocols for ad-hoc networks," *IET communications*, vol. 2, no. 5, pp. 650–657, 2008.

[12] A. Loch, M. Hollick, and A. Kuehne, "Practical ofdma for corridor-based routing in wireless multihop networks," *Proc. LCN 2014*, pp. 1–9, 2014.

[13] K. Bao, F. Hu, E. Bentley, and S. Kumar, "Diamond-shaped mesh network routing with cross-layer design to explore the benefits of multi-beam smart antennas," *International Conference on Computer Communications and Networks (ICCCN), WiLAN workshop, 2016, Waikoloa, Hawaii, USA, August 1-4,*, 2016.

[14] P. Lauder, R. Kummerfeld, and A. Fekete, "Hierarchical network routing," in *Communications Software, 1991,'Communications for Distributed Applications and Systems', Proceedings of TRICOMM'91., IEEE Conference on*. IEEE, 1991, pp. 105–114.

[15] L. Kleinrock and F. Kamoun, "Hierarchical routing for large networks performance evaluation and optimization," *Computer Networks (1976)*, vol. 1, no. 3, pp. 155–174, 1977.

[16] D. A. Nugroho, A. Prasetiadi, and D.-S. Kim, "Male-silkmoth-inspired routing algorithm for large-scale wireless mesh networks," *Journal of Communications and Networks*, vol. 17, no. 4, pp. 384–393, 2015.

[17] L. F. M. Vieira, U. Lee, and M. Gerla, "Phero-trail: a bio-inspired location service for mobile underwater sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 4, pp. 553–563, 2010.

[18] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," *Ad hoc networks*, vol. 1, no. 2, pp. 293–315, 2003.

[19] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer networks*, vol. 38, no. 4, pp. 393–422, 2002.

[20] B. Yuan, M. Orlowska, and S. Sadiq, "On the optimal robot routing problem in wireless sensor networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 9, pp. 1252–1261, 2007.

[21] P. Basu and J. Redi, "Movement control algorithms for realization of fault-tolerant ad hoc robot networks," *IEEE network*, vol. 18, no. 4, pp. 36–44, 2004.

[22] N. Michael, M. M. Zavlanos, V. Kumar, and G. J. Pappas, "Maintaining connectivity in mobile robot networks," in *Experimental Robotics*. Springer, 2009, pp. 117–126.

[23] M. M. Nelson and W. T. Illingworth, *A practical guide to neural nets*. Addison-Wesley Reading, MA, 1991, vol. 1.

[24] O. Sporns, "Contributions and challenges for network models in cognitive neuroscience," *Nature neuroscience*, vol. 17, no. 5, pp. 652–660, 2014.

[25] B.-N. Cheng, M. Yuksel, and S. Kalyanaraman, "Orthogonal rendezvous routing protocol for wireless mesh networks," *IEEE/ACM Transactions on Networking (ToN)*, vol. 17, no. 2, pp. 542–555, 2009.

[26] D. Braginsky and D. Estrin, "Rumor routing algorthim for sensor networks," in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*. ACM, 2002, pp. 22–31.

[27] N. Javaid, A. Javaid, I. A. Khan, and K. Djouani, "Performance study of etx based wireless routing metrics," pp. 1–7, 2009.

[28] A. Rao and I. Stoica, "An overlay mac layer for 802.11 networks," in *Proceedings of the 3rd international conference on Mobile systems, applications, and services*. ACM, 2005, pp. 135–148.

[29] W. Zhou, X. Chen, and D. Qiao, "Practical routing and channel assignment scheme for mesh networks with directional antennas," in *2008 IEEE International Conference on Communications*. IEEE, 2008, pp. 3181–3187.