University of Victoria

Review of Modern Issues for Unmanned Aerial Vehicle Swarm Communication and Management.

Dafydd Foster
Department of Computer Science
Dafyddfoster@gmail.com

Paige Loffler
Department of Computer Science
Paige.lofflerr@gmail.com

Jacob Thom
Department of Computer Science
Thejacobthom@gmail.com

Abstract

Unmanned Aerial Vehicles (UAVs) have become ever more pervasive in modern lives, applicable to military operation, commercial applications and civilian use. It has become increasingly more common for UAVs to be formed into swarms in order to carry out tasks quickly, efficiently, and with added redundancy should any of the UAVs be forced out of service. The added complexity that is involved in managing a collection of UAVs comes with the inherent issues found in Ad-hoc networks. These issues are further added to by the highly dynamic nature of UAV swarms. This review article focuses on three cornerstone issues and their modern-day solutions which have been developed within the last half decade.

1. Introduction

FANET is a promising field which offers a number of advantages over current network infrastructures. Due to the inherent speed that is associated with UAVs, FANET allows superior mobility compared to alternative structures. Costs may also be kept minimal due to its headless and/or remote nature. While there are numerous advantages, it is not fit to say that FANET is without disadvantages and issues.The issues that are most prevalent in the application of UAV swarms are maintaining low latency between nodes, organizing nodes into an efficient physical topology, and ensuring the nodes have the required energy to complete their mission. The stated concerns may be considered the most pertinent for FANET since without careful management of these three points, a UAV swarm fails to exist within a useful limit.

Low latency is required between nodes in order for effective communication and coordination to be completed in a highly dynamic environment. As connection speed is lowered, so too is coordination speed, increasing the risk for UAV swarms to malfunction or fail to complete their mission. Overhead is intrinsically a part of controlling for latency, either overhead which results in latency between nodes increasing, or overhead that aims at reducing latency. Without network overhead there can be no network and as such, no network latency. Thus, the problem becomes controlling overhead so that latency is reduced between nodes in an efficient manner.

Topology management is incredibly important in order to maximize network coverage and area coverage for the applications of the UAVs. The main difficulty regarding FANET topology relates to initial relay node positioning. Due to the complexity of this issue it is known to be an np-hard problem and thus is computationally expensive. Solutions have shown up to a 39 minute execution time, which is far from optimal and results in time spent organizing an efficient physical topology which could otherwise be used for the mission.

UAV swarms by nature depend on effective energy management. Without efficient management and transmission, UAV network lifetime and feasibility would be stunted by poor resource availability. This notion is carried forward to application feasibility, which depends on energy being managed efficiently not only to complete mission tasks, but to maintain the flight time required to be of service for completing these tasks. This poses a problem as UAVs are generally energy-restricted devices to allow for greater mobility, yet they require more resources than mobile devices on the ground to support flight.

This paper is broken into four sections, including a conclusion. Section one discusses the previously mentioned latency problem, section two focuses on physical topology management, and section three focuses on energy efficiency.

2. Latency and Overhead

A common issue with FANET is the need to reduce latency, while at the same time reducing overhead. This is a challenge for many protocols for FANET specifically due to their dynamic topology nature. The network must be able to adapt and control for link failure, nodes connecting and disconnecting, along with the typical network issues such as traffic congestion.

For the purposes of this survey, we have elected to ignore static protocols, since these are by name not suited for dynamic topology networking. For each type of routing protocol (proactive, reacting, gps, SI) we looked at how different protocols seek to control latency and how this affects the overhead. The overhead required for a network is important for several reasons. Increased network overhead comes with an increase in network traffic specifically designed to control for and monitor the network topology. The increased network traffic in turn often causes increased congestion and energy - congestion consisting of its usual issues (lower bandwidth, dropped packets, increased latency) and energy being a coveted and valuable resource for UAVs.

2.1 Proactive Protocols

Proactive protocols control for latency by proactively seeking out routes for a network. This type of protocol proactively seeks routes and controls for new/lost nodes by periodically checking the network for changes to topology and maintaining routing tables to every node in the network. This reduces latency by requiring minimal time to find a route after the routing tables have been populated. This reduction in latency is counterbalanced by the increased network overhead, as these protocols are continuously monitoring the network for changes to topology. As of the time of writing, there are presently two routing protocols with widespread popularity, DSDV and

OLSR, along with their variations. This is not to say these protocols are the only ones currently in use or that they are indeed the best. We will focus on OLSR for the purposes of this paper.

OLSR, or Optimized Link State Routing, focuses on using HELLO and Topology Control (TC) messages to discover and analyze surrounding nodes. The information is used to create a model of the topology in order to compute next hop destinations for all nodes in the networking using the shortest hop paths. OLSR-LD [1] is a modern variation that focuses on the Quality of Service (QoS) of links when computing the next hop destination in the forwarding paths. QoS is a measure of how many packets are successfully transmitted versus those that are dropped. This is important for FANETs due to their high mobility causing nodes to quickly enter and leave the communication range of other nodes in the network. The protocol accomplishes this focus by storing the link quality of every node within a link cache that is then utilized when computing the shortest path, a path by this method may choose a longer path than what is theoretically the shortest path in order to maintain link quality. Monitoring for the QoS among nodes is important in reducing overhead for an algorithm that is already by definition, overhead intensive. By not just proactively looking for paths, but proactively looking for high quality paths, the algorithm reduces overhead that would otherwise come as a result of dropped packets from broken links. When compared to base OLSR, the end-to-end delay, otherwise known as latency, is reduced and the packet delivery ratio is increased.

2.2 Reactive Protocols

Contrary to proactive protocols which focus on gathering and maintaining a full routing table for the entire network topology, reactive protocols wait until a node needs to communicate to a previously uncontacted node before it searches for a route. This reactive approach to route discovery leads to reduced network overhead by only sending out networking packets when specifically required. This leads to increased latency at two steps: (1) If a route between two nodes does not yet exist within a routing table it must find a route, which takes time and therefore increases latency; (2) if a route exists within the routing table, latency can occur if the route is broken or otherwise unavailable, causing the network to search for a new route after failing to transmit the initial packets. The following are examples of reactive protocols:

[1] Ad-hoc On-Demand Vector routing, or AODV as its commonly abbreviated, is a protocol that focuses on finding routes with the least latency by using minimal overhead. This is accomplished through the use of Route Request (RREQ) and Route Reply (RREP) messages, with another message, Route Error (RERR), to report and control for routing errors. As routes are reactively gathered they are stored in a routing hop table, which is reactively added to through monitoring RREP packets, and maintained by monitoring RERR Packets. At the present moment there are multiple variations for AODV, we chose to focus on the EV-AODV protocol developed by Wu, Shi, Liu, and Gu [2]. Their proposed protocol seeks to overcome the issues common with FANET - high node mobility, highly dynamic topology, and unstable link connections. The algorithm seeks to control for link stability by reviewing the current energy levels and relative speed of the link's nodes to evaluate its stability through the developed EV equation. On sending out RREQ packets, EV-AODV will update the path stability value at every

node until the packets reach their destination. Upon receiving the RREPs, the source node will select the path with the lowest EV value, contrasting AODV which selects the first path to successfully return an RREP. By selecting the most stable link as judged by the EV equation, the researchers found that their improvements led to reduced energy usage (increasing the overall lifetime of the network) as well as most importantly, reducing the amount of dropped packets as a result of previously unstable links when compared to AODV. The reduction in dropped packets reduces the overhead required to find new paths between nodes and thus decreases the latency between nodes in the network.

[2] Dynamic Source Routing (DSR) is another common reactive protocol. It is similar to AODV in that when a route does not exist in a cache, it requests a route through RREP and receives one through RREP if one exists. The difference lies in that routes do not expire in DSR, and are kept until proven broken, and that the entire routing path from source node to destination node is kept in the source node, rather than AODV which just records the next hop from node to node. In comparison to AODV, DSR reduces overhead even further at the cost of frequently keeping hold and attempting to use broken routes whereas AODV keeps a somewhat frequent track of next hops to take. Yang and Liu [3] have developed an optimization for DSR that improves the latency and packet-drop ratio that is commonly found with decreased overhead. Their algorithm, CHNN-DSR improves base DSR by utilizing a Hopfield neural network that takes in various environmental vectors and maintains routes through invalidation and confirmation. Their improvement by using a neural net causes consistent results for decreasing latency and increasing throughput of base DSR while maintaining relatively the same overhead.

2.3 Geographic/Position Based Protocols

Geographic/position based protocols work by sending packets and routing information to geographic addresses rather than network addresses. What is meant by this is that packets are sent to a physical location in the network, rather than a prescribed address. This serves to reduce overhead by often lacking the need for network routing by having packets sent based on relative physical location instead of a network topology. While network topology may be constantly shifting (thus requiring new routing tables for topology based protocols), geographic/position based protocols are able to send packets to the nearest neighbour of a location by greedy forwarding. This can induce routing loops, which is why greedy forwarding is typically paired with an intermediary step when a loop is a nearing possibility, such as face routing. The reduction in overhead and lack of network-addressed packets cause geographic/position based protocols to be inherently faster than topology based protocols (such as reactive and proactive) at the cost of address-based communication.

An example of a geographic/position based protocol would be GPSR, which greedily forwards packets towards a geographic location and forwards packets using perimeter mode when no such path exists from the current node. Issues that exist with other FANET protocols still exist in GPSR, a recent variation of this protocol, TQNGPSR [4] seeks to overcome these issues by employing traffic-awareness and a Q-network to determine optimal paths. TQNGPSR works on several levels to fix pervasive issues in geographic/position based routing. The predecessor of TQNGPSR, QNGPSR [5](also by Lyu et. al.) obtains a visit list from every packet so that it is actively aware of where packets came from geographically in order to lower dependency on perimeter mode. As well as a visit list, data for the Q-network is gathered in the form of neighbour topology through HELLO and HELLO-ACK messages. Where TQNGSPR differs from its predecessor is in gaining the length of the queue tables for its neighbours and adding them as a vector through which to calculate the Q-values from the Q-network. By actively monitoring link state, neighbour topology, and neighbour queues, TQNGSPR maintains a relatively minimal overhead that reduces congestion as well as actively controlling for congestion. Research has shown that as the number of nodes in a network increases,TQNGPSR outperforms the previously mentioned OLSR and AODV, as well as maintaining end-to-end delay at the lowest bound and only being outperformed by base GPSR in terms of overhead size.

2.4 Swarm Intelligence Based Protocols

Swarm Intelligence (SI) based protocols are protocols that seek to mimic existing ‘multi-agent’ swarms that can be found in the wild, such as fish, birds, bees, etc. By mimicking pre-existing swarming techniques, they hope to gain optimization that may have come with millions of years of evolution. Due to these protocols being based on a wide variety of sources, it would be an injustice to judge them all at once by blanketing statements. We will look at two existing SI based protocols and analyze them individually.

[1] BeeAdHoc [6] is a bio-inspired swarm intelligence protocol that attempts utilizing packets to mimic forager bees in two stages of routing. When a route is not yet available, BeeAdHoc reactively enters a stage of routing called “scouting” in which the source node sends out multiple forward-scouting packets that analyze the energy required, hop count, visit-list, etc., at every node along multiple paths until they reach the destination node. Upon reaching the destination node, backward-scouts are sent to the source node along the discovered paths, which are added to a routing table. After scouting has been completed, data begins transmitting among the paths stored in the table. Using a complicated algorithm, the distribution of packets is varied by probability among the paths, with inefficient or broken paths being removed from the routing table. Due to the reactive and somewhat network flooding routing method, BeeAdHoc maintains an overhead and overall latency larger than other reactive protocols such as the previously mentioned AODV and DSR. The goal of this protocol as judged by the research results is not to decrease latency, but to increase throughput, a goal that it accomplishes with the sacrifice of latency and overhead-induced network congestion. Overall, this protocol fails to reduce both latency and/or overhead.

[2] African Buffalo Optimization (ABO) of AODV (B-AODV) [7] seeks to mimic african buffalos and their communication when pasturing. This is done through two methods consecutively after each other. When base AODV sends out RREQ messages ABO is run simultaneously. Under ABO, links possessing a high QoS emit a “maa” token and all sub-par paths emit a “waa” token. Sub-par paths are ignored and not returned with RREP. The RREP is further modified to contain hop-count and the energy required for that path in question. These variables alongside delay are compared against other paths that received RREPs with the final and optimal path being chosen from these three variables. These additions to the AODV protocol increase the overall network overhead but also serve to decrease the latency, showcasing the general expected results of increased network overhead.

3. Topology

While several of the papers reviewed for this survey discuss mission critical characteristics, the following discussion is limited to those sections pertaining to Network Topology Construction and Management.

A typical FANET topology consists of

- Ground control station (GCS): a Static control body
- Cluster: a group of UAVs which are interconnected
- Custer node (CN): an individual UAV which is part of a cluster.
- Custer head (CH): an UAV with the job of maintaining connectivity between a GCS and the rest of the cluster
- Relay node (RN): UAVs which job is to relay data between cluster nodes (CN's)

In the field of FANET, network topology can be broken down into two sections: Topology Construction, and Topology Management. Topology construction relates to the initial location assignments of UAVs while topology management is concerned about real time location management in attempts to keep an optimal network configuration following the initial construction.

There are a number of variables that the various algorithms attempt to optimize in both the construction and management algorithms, including: reducing delay, maximizing coverage, minimizing connection loss [8]. As such, there is no single solution metric but rather the success of a given algorithm depends upon the definition of optimality. A common definition of optimality is to have a k-connected network with a minimum number of relay nodes (RN’s) [9].

3.1 Network Topology Construction

FANET topology construction attempts to derive optimal initial locations for the UAVs. Since the mission UAV’s locations are directly related to the mission goal, the construction phase mainly attempts to determine optimal relay node location along with selection of the cluster head.

3.1.1 Relay Node Positioning

The problem of positioning the relay nodes is a very challenging issue. In fact, it is known to be an np-hard problem, and thus computationally expensive. The Particle Swarm Optimization (PSO) algorithm [10] is the main technique used in solving this issue.

Kim and Lee [11, 12] both use modifications of PSO. Kim and Lee [12] use a penalizing concept with penalties representing ‘end-to-end communication constraint’ and ‘safety constraint’. After each iteration of the algorithm the best-seen-so-far position is the position which results in the smallest penalty. Through simulation run on ‘MATLAB software on a computer with intel core i7-7700 cpu (3.60GHz) and 16.0 GB RAM’ it was shown that their modified PSO based algorithm had on average approximately 20 times faster execution time then standard PSO, 118 seconds compared to 2,370 seconds respectively. Kim and Lee [11] also use a penalizing approach where network and mission critical characteristics are balanced. Although focused on Mobile Adhoc networks (MANET), Magán-Carríon et al., [9] look in detail at the relay node positioning problem.

3.1.2 Particle Swarm Algorithm

The Particle Swarm Algorithm (PSO) is a metaheuristic algorithm based on swarm behavior seen in nature such as a flock of birds, or school of fish [10]. The benefits of this algorithm are that it is simple to implement and that it's accurate. The downsides to this method are that it is computationally intensive, resulting in loss of mission time.

Each particle's next location is determined by the particles current location and the velocity term, *equation 1*. This velocity term is made up of the particle's current position, its own best position, and the global best position as seen in *equation 2.*

$$X^{t+1}{}_{i}=X^{t}{}_{i}+V^{t+1}{}_{i}$$

Equation 1

$$V^{t+1}{}_{i}=w(V^{t}{}_{i})+(c_1)(r^{t}{}_{1})(pbest_i-X^{t}{}_{i})+(c_2)(r^{t}{}_{2})(gbest-X^{t}{}_{i})$$

Equation 2

A quick overview of these equations is as follows:

> *t* is equal to the iteration count, *i* equals the current particle, *pbest* equals the best position seen by particle *i* , gbest equals the best position seen by the entire cluster. *w* is the inertia weight constant and is a positive constant value. The value of *w* represents how much influence the particles previous motion has in regards to its next motion. *c1* and *c2* have the same effect as *w* does but in regards to particles i's previous experiences and the clusters global experience respectively. *r1* and *r2* are random values with range [0,1]. The purpose of these random variables is to stop premature convergence.

3.1.3 Cluster Head Assignment

In FANET Topology, Cluster Head (CH) refers to one specific UAV which is in charge of maintaining connection between the ground control station (GCS) and the cluster the CH belongs to. Typically, energy and connection levels of each UAV are compared in the selection of the optimal CH.

Khan et al., [13] use the Glowworm Swarm Optimization (GSO) algorithm to select a cluster head. This categorizes nodes based on fitness which is calculated by residual energy and luciferin levels. The UAV with the highest fitness rating is selected as CH.

Joshi et al., [14] use a similar approach in regards to classifying the UAVs into categories but it uses energy and connectivity levels as parameters. Depending on the result each UAV is placed into one of nine categories. UAVs that are classed as P1 (highest energy and connectivity levels) self assign themselves as a Temporary Cluster-Head (TCH). These TCH's then compare against each other and the best is selected as the Final Cluster-Head (FCH).

Park et al., [15] does an in depth analysis of three different CH selection techniques: Random, Maximum residual energy, and hybrid. The results found that the expected lifetime of the UAVs were best when using a hybrid selection technique. The selection parameters for their hybrid technique looked at two things, the UAVs residual energy and the distance between the ground control station and the UAV

3.2 Network Topology Management

Network management deals with the issue of a FANET being highly dynamic. Once the construction phase has completed, each UAV has an initial location. Due to the high dynamic levels seen in FANET these previous locations become sub optimal in minimal time steps. The management phase looks at correcting the topology in real time in an attempt to keep the network optimal. Since the mission UAV's location is mission dependent, this phase mainly deals with correcting the relay nodes position.

Kim and Lee [11, 12] use a gradient method to accomplish this. [11] uses gradient ascent where [12] uses gradient descent. This technique has the benefits of being computationally light, and easily decentralized. Each RN can calculate its gradient by the information given to it by its direct neighbors alone, and adjust its location accordingly. Typically, this method also uses a threshold. In an attempt to keep the network optimal, each RN will self adjust based on its gradient without recalculating the routing path. Once an RN reaches its threshold it assumes large enough changes in the topology has occurred and will update its routing path in search for a more optimal route.

Qi et al., [8] define a K-hop sub graph as: " given G is an induced subgraph of G0, if all vertices of G are exactly composed by vertex v, and all its vincinal nodes within a k-hop, G is called a k-hop induced subgraph of v and denoted by Gkv"

[8] attempts to optimize connectivity and reduce topology changes through graph theory.

Representing a cluster as an undirected graph, Qi et al., [8] attempt to efficiently, both in terms of communication delay and computation overhead, eliminate local critical nodes of a k-hop induced subgraph. This is accomplished by introducing augmentation edges with a simplified block-cut tree. The result of this process is the topology represented in a simple manner, excluding non-crucial information .

Khan et al., [13] proposes a different view of the issue. Khan et al., [13] proposes cluster node location maintenance is a byproduct of the CH's movement. Each CN uses the Krill Herd (KH) [16] algorithm to determine the motion to be taken through use of its memory of the previous and current location of the CH along with its own current motion.

4. Energy Efficiency

The concern with FANETs and energy efficiency due to the fact that they are built over energy-restricted devices [17]. A large number of factors need to be taken into consideration when designing energy saving solutions for FANETs. These factors include maintaining feasible network topology and connection stability, as well as controlling for the weight, size, and power consumption rates of the UAVs. At the time of writing there does not exist widely available research on the topic of energy efficiency within FANET networks. We will go over the three main current solutions to energy-efficiency related problems in FANETs and describe how each solution works to improve real world efficiency.

4.1 Reputation-Aware Energy-Efficient Solution for FANET monitoring

The Reputation-Aware Energy-Efficient (RAEE) solution provides a technique that establishes trust among peers and distributes the network monitoring between said trusted peers with like mobility patterns [17]. This approach is important as it requires less computational power than similar protocols, which saves more energy.

4.1.1 Solution Overview

There are two main modules in this solution's architecture: the trust establishment module, and the monitoring periods manager. The trust establishment module focuses on direct and indirect trust computation, and the monitoring periods manager focuses on monitoring duration estimation, which estimates the overhead involved in the monitoring period and determines when it should start. The communication between these modules is visualised in Figure 2.

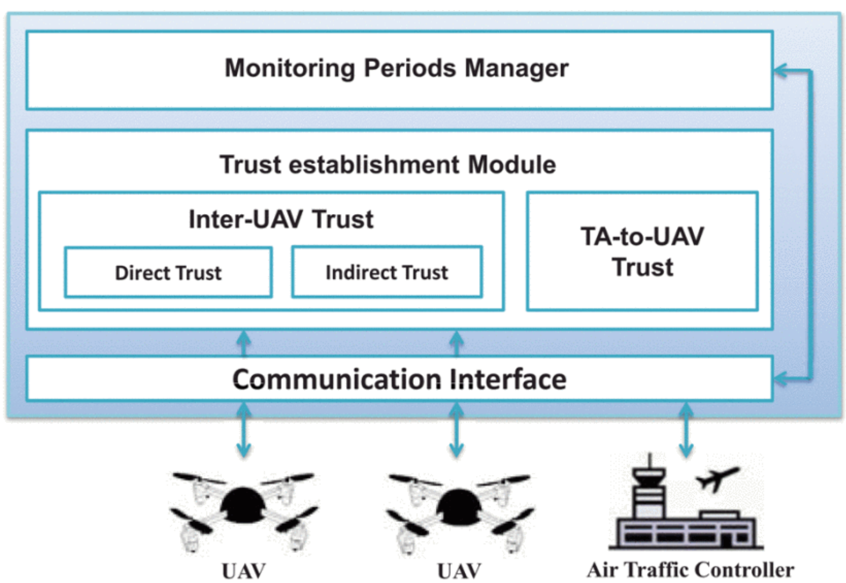


Figure 2. Architecture of the proposed solution [17].

To determine if one UAV has the trust of another UAV a Friend Score (FS) is used, which is computed with the following equation, *equation 3*, discussed in [17]:

$$FS(i, j) = T_{i,j} * LD(i, j)$$

Equation 3

Where $T_{i,j}$ is the trust between two nodes i and j, and *LD(i, j)* is the link duration between two nodes i and j. These two variables are used to represent the trust evaluation of two UAVs i and j. A node will be considered a friend when the FS surpasses a predetermined threshold. This determines if there exists both a high trust value and a stable communication medium between UAV nodes, displayed in Figure 3.

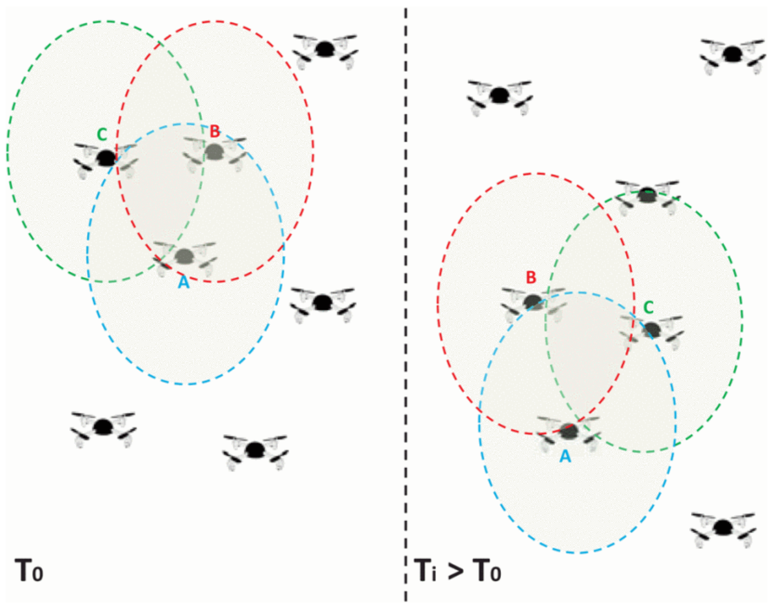


Figure 3. UAVs putting trust in one another and moving with a shared pattern [17].

Before an FS is determined, a direct and indirect trust score is evaluated. The direct trust score is defined as the local knowledge-based evaluation of the direct interactions among UAVs [17] and the indirect trust is the evaluation of the direct interactions between two UAVs based on the opinions of other UAVs about the honesty of the participating UAVs[17].

The solution proposed by RAEE begins with UAV nodes monitoring the network when they first become active. The task of monitoring the network is then distributed among nodes in the network as they begin to trust one another, resulting in the UAVs saving their own energy while the remainder of the nodes in the network take their turn of monitoring[17]. Once the trust table of each node has been fully updated the nodes gain the ability to trust their neighbours. After this stage has been reached nodes acquire the ability to share the monitoring process. If a monitor detects an untrustworthy node, as in if a neighbouring nodes trust value goes under the threshold, the monitoring node broadcasts a negative recommendation with the malicious node's identity in order for the other neighbouring nodes in the network to adjust their trust levels of the node in question.

4.1.2 Efficiency Results

In order to determine the efficiency of the proposed solutions, a set of simulations were run using Network Simulator - 2[17]. In the simulation, nodes were set to move about a grid map and their performance was gathered. Since the solution aimed to reduce energy consumption through the distribution of monitoring operations, a linear energy model was used to determine the cost, as seen in *equation 4*.

$$Cost = m * size + f$$

Equation 4

In equation 4, *m* represents the incremental cost, *size* represents packet size, and *f* represents the fixed costs, such as costs for accessing the channel. In reference to packet transmission, equation 5 is used to define the energy cost of each packet, which is actively monitored by the UAVs.

$$Cost_{pro} = \sum_{n \in S} f_{disc} + \sum_{n \in D} f_{disk} + \sum_{n \in S} (m_{pro} * size + f_{pro}) + \sum_{n \in D} f_{disc}$$

Equation 5

Where the set S includes all nodes in the transmission range of the sender node, the set D includes all nodes in the destination range, f_{disc} represents the cost of discarding control packets, m_{pro} represents the incremental cost of receiving a packet in promiscuous mode, and f_{pro} represents the fixed cost for each packet received in this mode.

A scenario with 20% of malicious UAVs was analyzed both with and without the distributed monitoring process in respect to network density. As you are able to see by the results in Figure 4, the solutions distributed monitoring technique was able to reduce average energy consumption by just over two times within high density cases[17].

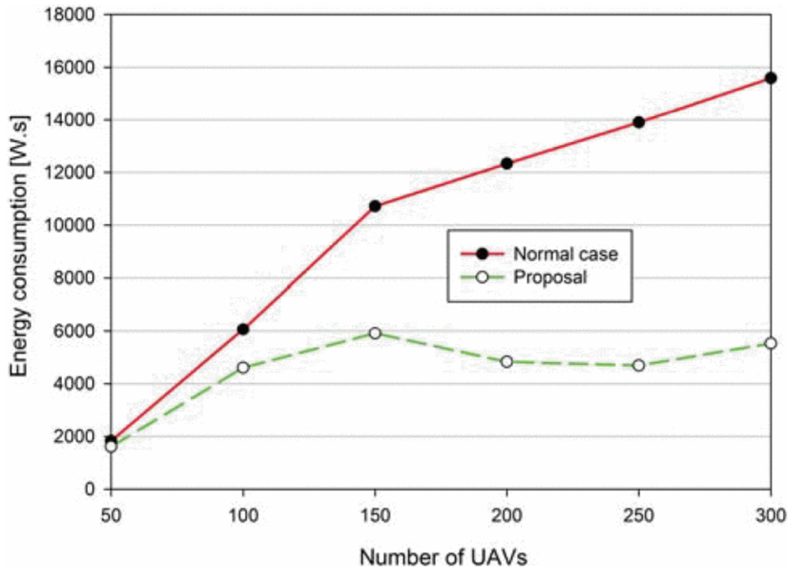


Figure 4. Proposal energy efficiency with 20% dishonest UAVs. [17].

This solution gives an overview on decreasing energy consumption within FANETs while still ensuring stable security levels.

4.2 Smart IoT Control-Based Nature Inspired Energy-Efficient Routing Protocol

The Smart IoT Control-Based Nature Inspired Energy-Efficient routing protocol presents a routing protocol based on a modified AntHocNet in order to take advantage of colony optimization and provide better dependability and performance[18].

4.2.1 Protocol Overview

The proposed solution is a protocol E-AntHocNet that is a nature inspired energy efficient routing scheme for FANET. The routing protocol is based around ant colony metaheuristics[23]. There are three steps within this protocol that include the generation of an initial feasible solution, the daemon action if any (optional), and the pheromone update process[18].

The first step of generating initial solutions begins with iteration. This protocol builds on the fact that nature inspired algorithms are iterative in nature and will follow a course to find the best solution, building a new feasible solution on each iteration[20]. The equation below details the probability of an ant selecting a next node j from node i:

$$P_{i,j}^{k} = \frac{T_{i,j}^{a} * n_{i,j}^{B}}{\Sigma_{l \in N_i} T_{i,j}^{a} * n_{i,j}^{B}}$$

Equation 6

Where $P^{k}_{i,j}$ is the probability of transition of the Kth ant from vertex i to j using link l, $T_{i,j}$ is the quantity of pheromone on the edge l, $n_{i,j}$ is heuristic value of link l which is normally $1/d_{i,j}$ where $d_{i,j}$ is the Euclidean distance between i and j, N_i represents neighbours of node i, and a and B are parameters to weigh the significance of pheromone and distance in the selection of the next vertex.

The next optional step involves the provoking Daemon actions to perform problem or scenario specific tasks[18], which are normally called once the solution construction iterations have completed.

The final step is the pheromone update process. This step is put in place to increase the value of $T_{i,j}$ for good solutions with reinforcement[18] in order to select a path.

4.2.2 Efficiency Results

Using the Network Simulator - 2 the eAntHocNet routing protocol was tested. The evaluation parameters that the protocol was tested against include the packet drop rate, network throughput, packet received ratio, average end-to-end delay, and work done.

The results achieved from the simulation show that the eAntHocNet had a larger chance of reducing the packet drop rate. Having a smaller amount of dropped packets increases the throughput of a network and the quality of service it provides. Compared to its base protocols AntHocNet, DSR, MDART, and TORA, eAntHocNet was more efficient in all aspects[18]. This boost in efficiency can be seen in Figure 5.

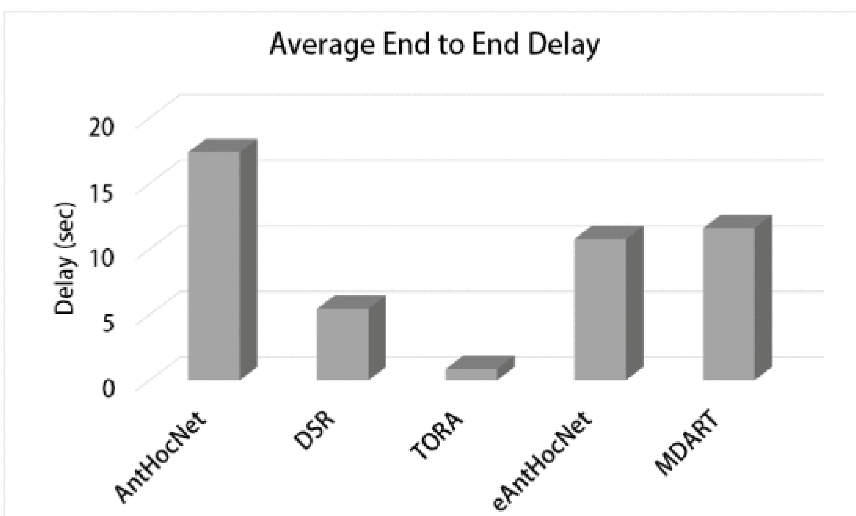


Figure 5. Average end-to-end delay of packet transmission. [18].

4.3 Energy-Efficient Connectivity-Aware Data Delivery Protocol (ECaD)

Energy-Efficient routing which takes into consideration movement information and energy levels of each UAV to guarantee stable communication between devices and link-breakage predictions, prior to them occurring[19].

4.3.1 Routing Overview

The main goals behind Energy-efficient connectivity-aware data delivery (ECaD) is to make the routing paths that are already in use stronger, building off of AODV, but improved in order to take advantage of control messages to achieve the ability to determine how connected the links between UAV nodes are, and the energy levels of said nodes in a FANET.

In order to prevent faults when routing, a more effective maintenance process is used so a node does not have to initialize a discovery process multiple times in the case of a disconnect, and instead provides different paths to maintain the connection[19]. To avoid further delays and traffic within a network ECaD uses static size routing packets. This enables the routing process to skip having to record full routing paths and limits overhead. ECaD stands out against typical routing protocols as it builds multiple paths that lead toward a destination[19]. The protocol uses three different processes to aid in creating, using, and maintaining a path between notes. These processes include the discovery process, maintenance process, and data delivery process.

The discovery process floods a route request packet over a FANET to discover the different path options it is able to take[19]. As the process does this it also takes into consideration the different UAV nodes willingness to participate in the process for other UAVs. This is decided based upon the residual energy within said node, as when the residual energy is higher than the threshold it will broadcast a signal, else drop it[19].

The maintenance process begins when a UAV detects a failure and wants to begin searching for a new solution to keep the connection. This ensures the routing protocol is able to continue to keep a stable connection between nodes that have a high mobility in FANETs[19]. Once a failure has been detected the UAV node’s routing table entry is considered to be invalid and gets removed from the routing table[19]. The UAV that detected the failure then refers back to its routing table to determine the next neighbour that is capable of forwarding the packet.

And finally the data delivery process ensures the source will begin to transmit data after receiving the RREP packet. This process remains similar to that of AODV where a routing table is checked before transmitting the data, but the routing tables update as data is being sent to ensure a stronger connection and less downtime[19].

These three processes are visible in Figure 5:

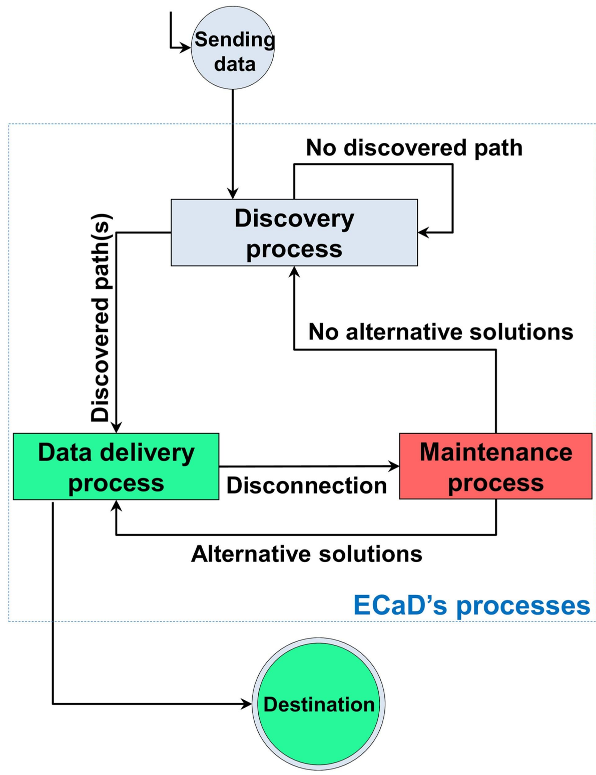
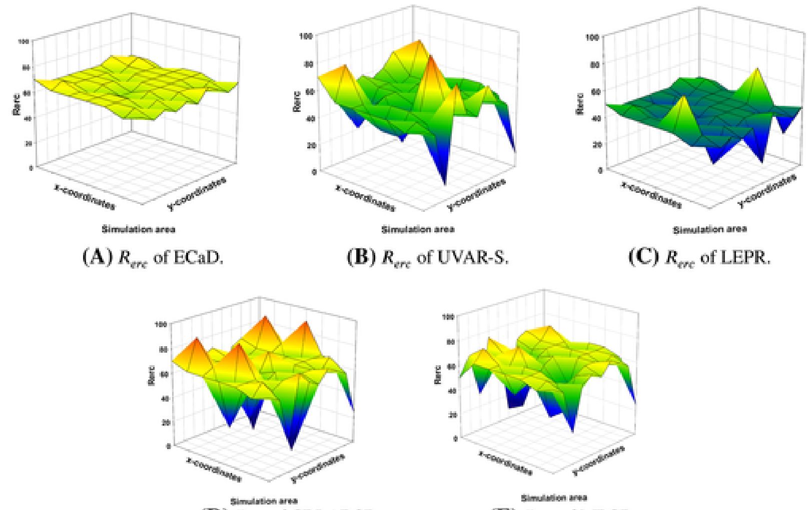


Figure 5. Processes of energy-efficient connectivity-aware data delivery (ECaD). [19].

4.3.2 Results

Using the Network Simulator - 2 the ECaD protocol was tested against similar protocols with the same parameters, these protocols include the UAV-Assisted VANET Routing protocol (UVAR-S), the Mobility Prediction-based Geographic Routing protocol (MPGR), the Link Stability Estimation-based Preemptive Routing protocol (LEPR), and the Cluster-Based Location-Aided Dynamic Source Routing protocol (CBLADSR).

The evaluation results can be split up into 5 different categories. The first being the ratio of energy remaining capacity. When investigating the contour of residual energy level of 49 UAVs that are uniformly deployed over a 2000x2000m area, it is possible to see that ECaD has a more balanced energy consumption compared to similar protocols, seen in Figure 6:



(A) R_{erc} of ECaD. (B) R_{erc} of UVAR-S. (C) R_{erc} of LEPR.

(D) R_{erc} of CBLADSR. (E) R_{erc} of MPGR.

Figure 6. Contours of residual energy levels of 100 unmanned aerial vehicles (UAVs) at the end of the simulation [19].

The second category involves the packet delivery ratio. After running the tests ECaD had increased the packet delivery ratio by 50% compared to the UVAR-S, LEPR, and MPGR protocols, and more than 10% compared with CBLADSR[19]. This occurrence is caused by the number of possible paths that are discovered between nodes due to ECaDs more efficient routing discovery process[19]. The other protocols included in the simulation suffered from a larger packet loss due to the lack of balance in energy consumption[19].

The third category involves end-to-end delay. The simulation determined that the average end-to-end delay of ECaD and MPGR was around 20% lower than the other protocols involved in the simulation[19]. The difference between ECaD's results and MPGR's results being that ECaD was able to achieve that result due to its large amount of paths that continue to be valid without needing to reinitialize another route discovery process, while MPGR achieved the results based off of the fact it is observed to be a more advantageous protocol in comparison to ECaD, using an enhanced version of a greedy forwarding routing protocol[21].

The fourth category is control overhead. For the simulation to calculate the overhead it calculates the total number of extra routing packets during each transmission of data and divides the value over the number of successfully received packets of data at the destination. The simulation found that ECaD's overhead was lower than UVAR-S and LEPR in 80% of cases tested[19]. Similar to end-to-end delay, the reasoning on why ECaD had a much lower overhead is because of the protocol's ability to upkeep its routing paths due to the protocol's connectedness and energy efficiency[19]. However, there exists an even smaller overhead within the CBLADSR and MPGR protocols due to their exchanges of hello packets and the clusters they form[22].

And lastly the fifth category is path stability and data reliability. Compared to other protocols ECaD takes into consideration mobility, energy consumption, and delivery delay to find the best path. Since the paths chosen by ECaD had longer durations, less path breaks, and an awareness of movement it was able to achieve better data reliability compared to other protocols[19].

Based off of the simulations ECaD was able to offer a more reliable, energy saving routing protocol for UAVs within an FANET compared to other routing protocols, the only negative aspect to using this protocol in opposed to protocols such as CBLADSR and MPGR would be a larger overhead[19].

5. Conclusion

As was shown to us through the course of our survey, the field of research surrounding UAV swarm networks is not without issues. These issues present numerous possibilities for solutions, some often working more effectively than others. As seen through the presented networking protocols, the inverse relationship between increased network overhead and lowered latency isn't always true. It can be stated as a rule of thumb that overhead which seeks to monitor the link state between nodes and traffic can cause a reduction in latency by selectively choosing routes that produce optimal long-term end-to-end delay. When looking at current solutions being researched for topology creation and management, advancements are seen in regards to execution time for the relay node positioning problem, simplification of topology representation though graph theory, and with techniques for optimal cluster head assignment. These presented solutions are not without issues of their own, solutions in problem spaces sometimes requiring sacrifices in one facet of the issue in order to improve other facets, leading to comparisons being easily made with no single specific strategy for developing a solution. In terms of energy efficiency, currently there does not exist an extensive list of resources for FANETs. While a newer field of research, solutions have still been developed for the issue of energy management in UAV network structures and leave plenty of room for research and improvement. All issues with their current respective solutions present FANET as a highly active research field, seeing constant new development at both the physical and network management levels. While still being in its infancy, it is doubtless that UAV swarm management techniques will be improved upon and as such possess great potential to redefine many aspects of our lives

References

[1] R. Jain and I. Kashyap, “An QoS aware link defined OLSR (LD-OLSR) routing protocol for MANETS,” *Wireless Personal Communications*, vol. 108, pp. 1745-1758, May 2019. [Online]. Available: Springer Link, https://www.link.springer.com. [Accessed: Dec. 04, 2020].

[2] J. Wu, S. Shi, Z. Liu, and X. Gu, “Optimization of AODV routing protocol in UAV ad hoc network,” in *Proc. of the First EAI International Conference on Artificial Intelligence for Communications and Networks, AICON 2019, 25-26 May 2019, Harbin, China* [Online]. Available: Springer Link, https://www.link.springer.com. [Accessed: Dec. 01, 2020].

[3] H. Yang and Z. Liu, “An optimization routing protocol for FANETs,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 120, pp. 1-8, May 2019. [Online]. Available: Springer Link, https://www.link.springer.com. [Accessed: Dec. 01, 2020].

[4] Y. Chen, N. Lyu, G. Song, B. Yang, and X. Jiang, “A traffic-aware Q-network enhanced routing protocol based on GPSR for unmanned aerial vehicle ad-hoc networks,” *Frontiers of Information Technology & Electronic Engineering*, vol. 21, pp. 1308-1320, Sept. 2020. [Online]. Available: Springer Link, https://www.link.springer.com. [Accessed: Nov. 25, 2020].

[5] N. Lyu, G. Song, B. Yang, and Y. Cheng, “QNGPSR: A Q-network enhanced geographic ad-hoc routing protocol based on GPSR,” in *Proc. of the 2018 IEEE 88th Vehicular Technology Conference, VTC-Fall, 27-30 Aug. 2018, Chicago, Illinois, USA* [Online]. Available: IEEE Xplore, https://www.ieee.org. [Accessed: Nov. 25, 2020].

[6] A. V. Leonov, “Modeling of bio-inspired algorithms AntHocNet and BeeAdHoc for flying ad hoc networks (FANETs),” in *Proc. of the 2016 13th International Scientific-Technical Conference on Actual Problems of Electronics Instrument Engineering, APEIE 2016, 3-6 Oct. 2016, Novosibirsk, Russia* [Online]. Available: IEEE Xplore, https://www.ieee.org. [Accessed: Nov. 28, 2020].

[7] M. H. Hassan, S. A. Mostafa, M. A. Mohammed, D. A. Ibrahim, B. A. Khalaf, and A. S. Al-Khaleefa, “Integrating African buffalo optimization algorithm in AODV routing protocol for improving the QoS of MANET,” *Journal of Southwest Jiaotong University*, vol. 54, no. 3, June 2019. [Online]. Available: Journal of Southwest Jiaotong University, https://jsju.org. [Accessed: Nov. 29, 2020].

[8] X. Qi, H. Jin, Q. Zhang, and Z. Yang, “A distributed bi-connectivity maintenance mechanism for flying ad hoc network topology”, in *Proc. of the IEEE/CIC International Conference on Communications in China, ICCC 2019, 11-13 Aug. 2019, Changchun,*

China [Online]. Available: IEEE Xplore, https://www.ieee.org. [Accessed: 06 Dec. 2020].

[9] R. Magán-Carrión, J. Camacho, P. García-Teodoro, E. F. Flushing, and G. A. Di Caro, "A dynamical relay node placement solution for MANETs," *Computer Communications*, vol. 114, pp. 36-50, Dec. 2017. [Online]. Available: ScienceDirect, https://www.science direct.com. [Accessed: Dec. 06, 2020].

[10] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. of the International Conference on Neural Networds, ICNN 1995, 27 Nov. – 1 Dec. 1995, Perth, Australia* [Online]. Available: IEEE Xplore, https://www.ieee.org. [Accessed: Dec. 06, 2020].

[11] D. Kim and J. Lee, "Joint mission assignment and topology management in the mission-critical FANET," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 2368-2385, March 2020. [Online]. Available: IEEE Xplore, https://www.ieee.org. [Accessed: Dec. 06, 2020].

[12] D. Kim and J. Lee, "Integrated topology management in flying ad hoc networks: Topology construction and adjustment," *IEEE Access*, vol. 6, pp. 61196-61211, 2018. [Online]. Available: IEEE Xplore, https://www.ieee.org. [Accessed: Dec. 06, 2020].

[13] A. Khan, F. Aftab, and Z. Zhang, "BICSF: Bio-inspired Clustering Scheme for FANETs," *IEEE Access*, vol. 7, pp. 31446-31456, 2019. [Online]. Available: IEEE Xplore, https://ieee.org. [Accessed: Dec. 06, 2020].

[14] A. Joshi, S, Dhongdi, S. Kumar, and K. R. Anupama, "Simulation of multi-UAV Ad hoc Network for disaster monitoring applications," in *Proc. of the 2020 International Conference on Information Networking, ICOIN 2020, 7-10 Jan. 2020, Barcelona, Spain* [Online]. Available: IEEE Xplore, https://www.ieee.org. [Accessed: Dec. 06, 2020].

[15] J. Park, S. Choi, H. R. Hussen, and J. Kim, "Analysis of dynamic cluster head selection for mission-oriented Flying Ad hoc Network," in *Proc. of the 2017 Ninth International Conference on Ubiquitous and Future Networks, ICUFN 2017, 4-7 July 2017, Milan, Italy* [Online]. Available: IEEE Xplore, https://www.ieee.org. [Accessed: Dec. 06, 2020].

[16] A. H. Gandomi and A. H. Alavi, "Krill herd: A new bio-inspired optimization algorithm," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 12, pp. 4831-4845, Dec. 2012. [Online]. Available: ScienceDirect, https://www.sciencedirect. com. [Accessed: Dec. 06, 2020].

[17] C. A. Kerrache, E. Barka, N. Lagraa, and A. Lakas, "Reputation-aware energy-efficient solution for FANET monitoring," in *Proc. of the 2017 10th IFIP Wireless and Mobile*

Networking Conference, WMNC 2017, 25-27 Sept. 2017, Valencia, Spain, [Online]. Available: IEEE Xplore, https://www.ieee.org. [Accessed: Dec. 05, 2020].

[18] I. U. Khan, I. M. Qureshi, M. A. Aziz, T.A. Cheema, and S. B. H. Shah, “Smart IoT control-based nature inspired energy efficient routing protocol for flying ad hoc network (FANET),” *IEEE Access*, vol. 8, pp. 56371-56378, March 2020. [Online]. Available: IEEE Xplore, https://www.ieee.org. [Accessed: Dec. 05, 2020].

[19] O. S. Oubbati, M. Mozaffari, N. Chaib, P. Lorenz, M. Atiquzzaman, and A. Jamalipour, “ECaD: Energy-efficient routing in flying ad hoc networks,” *International Journal of Communication System*, vol. 32, no. 8, e4156, Aug. 2019. [Online]. Available: Wiley Online Library, https://www.onlinelibrary-wiley.com. [Accessed: Dec. 05, 2020].

[20] X. Yang, “Analysis of algorithms,” in *Nature-inspired optimization algorithms*. London, UK: Elsevier, 2014, 23-44. [Online]. Available: ScienceDirect, https://www.science direct.com. [Accessed: Dec. 05, 2020].

[21] L. Lin, Q. Sun, S. Wang, and F. Yang, “A geographic mobility prediction routing protocol for ad hoc UAV network,” in *Proc. of the 2012 IEEE Globecom Workshops, 3-7 Dec. 2012, Anaheim, California, USA* [Online]. Available: IEEE Xplore, https://www.ieee.org. [Accessed: Dec. 05, 2020].

[22] F. Aadil, A. Raza, M. F. Khan, M. Maqsood, I. Mehmood, and S. Rho, “Energy aware cluster-based routing in flying ad-hoc networks,” *Sensors*, vol. 18, no. 5, p. 1413, May 2018. [Online]. Available: Directory of Open Access Journals, https://www.doaj.org. [Accessed: Dec. 05, 2020].

[23] A. Prakasam and N. Savarimuthu, “Metaheuristic algorithms and probabilistic behaviour: A comprehensive analysis of ant colony optimization and its variants,” *Artificial Intelligence Review*, vol. 45, pp. 97-130, Oct. 2015. [Online]. Available: Springer Link, https://www.link.springer.com. [Accessed: Dec. 05, 2020].