People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

Mohamed Khider university of Biskra

Faculty Of Exact Sciences, Natural And Life Sciences

Department Of Computer Science

Memory:

Presented To Obtain the Academic Master's Degree in

Computer Science

Specialty: Network and Technology Of Information And Communication

---

Optimizing Deployment And Operation Of UAV Under Predictable Conditions

---

By: Tahri Abde SSamad

Acknowledgment :

Primarily i would thank God for my being able to complete this thesis, And i would like to express my special thanks of gratitude to my supervisor Pr: **Abdel Malik Bachir** as well as Dr: **Sahraoui Somia** who give me the golden opportunity to do this wonderful thesis, Which also helped me in doing a lot of research and i came to know about so many things.

I am really thankful to them.

Secondly i would also like to thank my parents and friends who helped me a lot in finishing this thesis .

Thanks again to all who helped me.

TABLE OF CONTENT :

Abbreviations and notations:

OLSR	**Optimized Link State Routing**
DSDV	**DESTINATION SQUENSED DESTINCE VECTOR**
STAR	**SOURCE TREE ADAPTIVE ROUTING**
TBRPF	**Topology Dissemination Based On Reversed-Path Forwarding**
DSR	**Dynamic Source Routing**
AODV	**On Demand Distance Vector**
ZRP	**Zone Routing Protocol**
DTN	**Delay Tolerant Network**
ORPAH	**Opportunistic Routing Protocol In Dynamic Ad-Hoc Network**
FANET	**Flying Ad-Hoc Network**
UAV	**Unmanned Ariel Vehicle**
GCS	**Ground Control Station**
MANET	**Mobile Ad-Hoc Network**

VANET	**Vehicular Ad-Hoc Network**
NLOS	**Non Line Of Sight**
CH	**Cluster Head**
GS	**Ground Station**
GCS	**Ground Control Station**
GN	**Ground Node**
FN	**Flying Node**
SOSUS	**Sound Surveillance System**
WSN	**Wireless Sensor Network**
LAR	**Location Aided Routing**
LEACH	**Low Energy Adaptive Clustering Hierarchy**
QOS	**Quality Of Service**
SPIN	**Sensor Protocol For Information Via Negotiation**

General introduction

Data collection is an important task in wireless sensor network and this task has been guided for several years by the most investigated theme among researchers which is energy efficiency. Indeed, data collection was studied first in WSNs with one single Sink then with multiple Sinks to balance the energy consumption among the relay sensor nodes. Moreover, other data collection schemes have been proposed based on the use of mobile sinks. It has been argued in the literature that a mobile sink may improve the energy dissipation compared to a static one. Indeed, The drawbacks of using a static sink are well known. For instance, the nodes that are in the sink vicinity deplete their energy much earlier compared to the nodes located farther away from the sink due to higher data relaying load.

In this thesis, we argue for the use of the UAV as a flying Sink. Although it is clear that such sink improves load balancing among the nodes, it is an open question whether this also leads to improvements in fairness and number of collected packets. Thus, we aim to study those performance metrics in UAV-assisted WSNs.

Obviously, using a flying sink to collect data from a ground sensor network is a challenging task because of the link intermittence and the dynamicity of the network, particularly when the data types are videos.

The primary goal of this thesis is to minimize energy consumption in wireless sensor aided by UAVs as sink nodes using leach protocol in collecting video data.

Chapter 01 : Generalities On Flying Ad-Hoc Network

1.1 INTRODUCTION:

The recent progress of wireless technology has been witnessed in our daily life, particularly because of the vast availability of low-cost WI-FI radio interfaces and other devices like GPS, sensors, micro-embedded computers, etc. All these innovative devices have paved the path for the development of small intelligent flying vehicles, e.g., unmanned aerial vehicles (UAVS), leading to the creation of a new kind of network called flying ad hoc network (FANET). Since the introduction of FANET, different kinds of civilian and military applications have emerged, such as the coordination of rescue teams on the ground, border supervising, and autonomous tracking.

in addition, there are also many civilian applications such as agricultural and yards monitoring, discovering oil fields, and film-making[1].

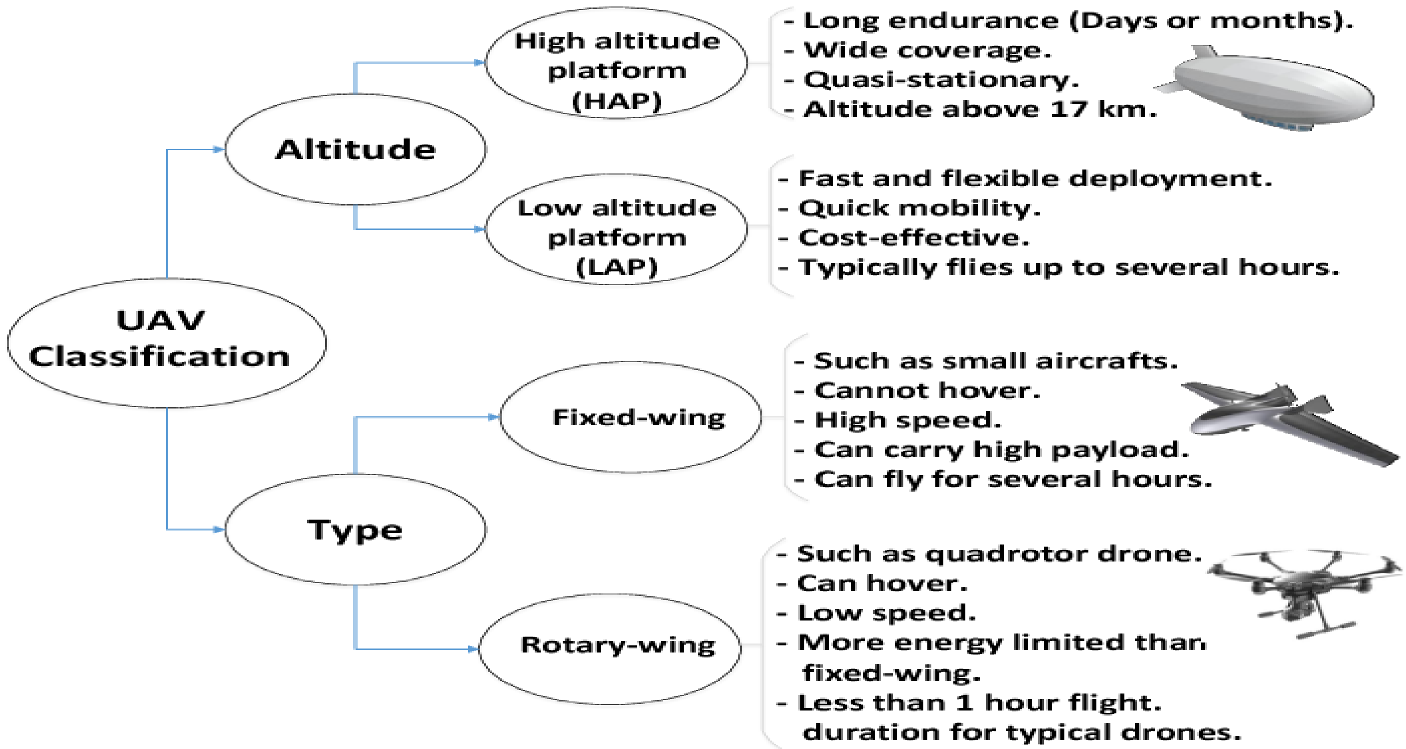

Fig. 1: UAV Classification.

FLYING AD-HOC NETWORK (FANET): [2]

is an ad-hoc network structure which is formed by a set of UAVS and at least one of them must be connected to ground control station **GCS** or satellite. **FANET** differs from existing ad-hoc networks, but it can be viewed as a special form of MANET or **VANET**. Though, there are certain differences between them in terms of design considerations. The UAVS network has its own characteristics, in the following point, some FANET characteristics are detailed .

1.2 FANETS CHARACTERISTICS :

as in MANETS, the FANET architecture is a network without infrastructure, using multiple nodes to transmit data packets. the other characteristics are self-organized abilities, self-managed data in a distributed manner, the nodes communications and cooperation that are used to deliver data. FANETS, however, have some specific features as well that makes them differ from MANET and VANET.

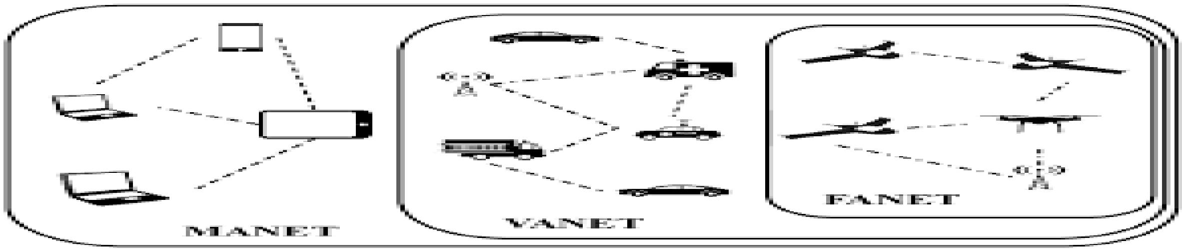

Fig 1.2: Ad-hoc Networks.

1.2.1 NETWORK CONNECTIVITY :

FANET network connectivity has higher intermittent degree than in MANETS or VANETS. it is caused by UAVS mobility. The communication interruption can be critical when important information (control/command traffic) is transmitted. Moreover, UAV failure may result in connectivity failure with a following routing failure, and subsequent communication failure or worse delay.

1.2.2 NODES NUMBER :

When a UAV with a relatively high velocity is deployed in a specific mission, a restricted mission area can be covered. in this case, a large number of UAVS is not necessary and a large number of nodes creates problems (deployment, interference, disaster …),in another wide area need a large number of nodes to be covered .

1.2.3 SUFFICIENT ENERGY :

it is usually supposed that depending on their size (there are different sizes for drones. the small ones are the most constrained) and type, FANET nodes have sufficient energy and computing power if compared to MANET nodes. The reason is that the amount of energy needed to move an UAV is much greater than that required to compute data and communication, and they are often densely deployed to overcome insufficient energy problems.

1.2.4 MOBILITY (3D) :

FANET mobility is high speed compared to other ad-hoc networks and, UAV can move in three dimensions.

1.2.5 ENVIRONMENT :

Since FANETS consist of flying nodes, they are usually used in large free spaces. Therefore, to model the physical layer one often uses the free-space path loss model. Still large obstacles, ground reflections or weather conditions and other factors can affect UAVS connectivity. These factors should be taken into consideration in the applied propagation model.

1.2.6 DELAY RESTRICTION :

As a rule, FANETS are employed in real-time applications, for instance aerial photography and video capture. Therefore, the control or command traffic should arrive timely, and in order to avoid loss of control, UAVS are to compute it with small delay,(delay might make disaster among UAVS if it is long).

1.3 FANET DESIGN:

Frequent topology changes result from high mobility. These must be considered when designing FANETS. other factors that require redefinition of routes include environmental obstacles such as mountains, failures of existing UAVS in the network, injection of new UAVS, possible mission updates, operational requirements, with taken into consideration adaptability , scalability, and latency, all of these constrains make the FANET design is difficult issue.

1.4 COMMUNICATION IN FANETS:

Communication is one of the most crucial design issues for FANET, thus, it is essential to define a reliable communication protocol for the UAVS networks (communication outage possibility is increased by high mobility).

Multiple UAVS or between UAVS. As shown in fig.3, communication architectures are introduced for UAV networks, namely, UAV direct communication (picture a), UAV communication via satellite networks (picture b), UAV communication via cellular networks (picture c) and UAV communication via ad-hoc networks (picture d).

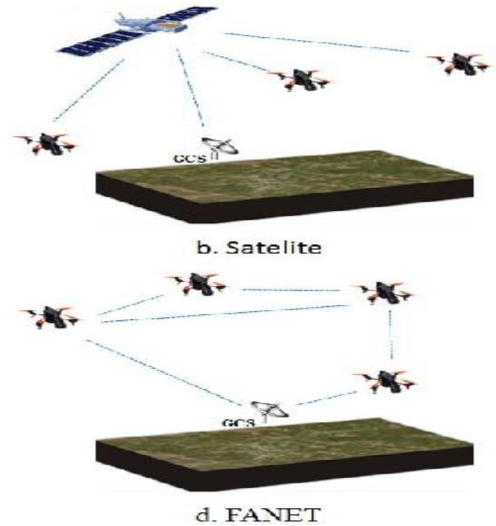

Fig 1.3: Basic FANETS Communication Architectures.

1.4.1 UAV DIRECT COMMUNICATION

A direct communication link can be used between the GCS and each UAV, as shown inFig.3(a).This is the simplest architecture, in which the GCS acts as a central node to which all UAVs are connected. However, the direct UAV to UAV communication is not possible because of the centralized scheme. This architecture cannot be applied for NLOS (Non-Line-Of-Sight) communication and dynamic environments. Moreover, the GCS represents vulnerability of FANET, for example if the GCS encounters some problems, the whole UAV network will fail.

1.4.2 UAV COMMUNICATION VIA SATELLITE NETWORKS :

In order to communicate two very distant points, located in regions where it is impossible to have a fixed infrastructure, satellite communications are the best solution, as shown in Fig.3(b). Satellites can provide communication between the GCS and the UAV in single UAV systems. Hence, for multi-UAVs systems, each aircraft can communicate with the GCS through a satellite. Thus, communications between UAVs can also pass in the same way. However, this approach has some weak points such as the high latency of transmission and the cost of leasing the satellite. In addition, the UAVs and the GCS must be on the satellite line of sight. Indeed, for some missions, trees or buildings can be obstacles against the signal exchanged between UAVs and their relay satellite. Moreover, the performance of satellite systems is related to the transmit power of the ground transmitters, which can be a disadvantage for mini or micro UAVs that are equipped with batteries with low capacity.

1.4.3 UAV COMMUNICATION VIA CELLULAR NETWORKS :

Nowadays the most used type of communication is the cellular network as illustrated inFig.3(c). Based on a centralized topology, this technology consists of cutting a territory into zones (cells), each of which is served by a base station (the central point). All communications must go through this central point which has the role of routing them to their destinations. Cellular communications is the foundation of mobile telephone technologies such as GSM, GPRS, UMTS, LTE and wireless data communications such as WIFI and WIMAX. Since they offer extreme freedom for nomadic users. Unlike satellite networks, cellular networks use low power transmitters. For this, it can be a solution to communicate the UAVs using the already existing telephone operator infrastructure to eliminate the constraints of the range and the mobility. However, the cost of communication is not negligible, even with the installation of a new infrastructure. In addition, it is difficult to cover all areas and keep this infrastructure safe in some cases, after natural disasters for example. Several researchers are working on the integration of UAVs in cellular networks.

1.4.4 UAV COMMUNICATION VIA AD-HOC NETWORKS :

In order to address the disadvantage of the communication architectures discussed above, the FANET network is proposed for a swarm of UAVs, as shown in Fig.3(d). This network architecture is a part of the MANET in which nodes communicate between them without the need for a central infrastructure. Each UAV is considered as an end system. All UAVs are required to collaborate and thus have to organize themselves to relay information. The Ad-Hoc architecture cope well with the constant changing topology of the UAV networks, that results from UAV high mobility. In FANET, the GCS acts also as a regular end node which can have a fixed or variable geographic allocation. It communicates with the nearest UAV which acts as a gateway. Therefore, there are two types of communication to consider in FANET: UAV to UAV communication and UAV to GCS communication.

1.4.4.1 UAV TO UAV COMMUNICATION:

UAVs communicate with each other in order to carry out common mission as cooperative target tracking or path planning. This type of communication can be direct or multi hop. UAVs can have short or long range of communication between them, and this can increase the efficiency of FANET in terms of communication range and data rate.

1.4.4.2 UAV TO GCS COMMUNICATION :

In this type of communication, UAVs communicate with fixed infrastructure (ground station, warship, cellular infrastructure, or satellite) to give information services for other users in the global network.

1.5 ROUTING PROTOCOLS IN FANETS:[2]

Several routing protocols were proposed in the literature for the Ad-Hoc networks, like, dynamic source routing, pre-computed routing, on demand routing, flooding, cluster based routing, etc. Because of FANET is a subclass of MANET and VANET, researchers firstly have tested protocols utilized in those networks for possible application in UAVs networks. However, due to the specific characteristics of nodes in FANET like speed, shortage of energy and rapid changes in links between them, most of these protocols cannot be directly used. Thus, it is important to modify these protocols to fulfill FANET requirements, nowadays we distinguish five routing protocols classification witch are:

STATIC ROUTING PROTOCOLS:

In this type of routing protocol, static routing tables are used which are computed and loaded when the task starts. Thus, there is no need to refresh or update these tables during an operation. Here every UAV communicates with a limited numbers of flying nodes or GCS , and it only preserves their information. In case of a failure (of an UAV or a GCS), for updating the tables, it is essential to wait until the assignment is finished. As result, these protocols are not fault tolerant and they are not suitable for dynamically changing network. Therefore, their applicability to real FANET networks is limited. These protocols are mainly :

LCAD(LOAD CARRY AND DELIVER) :

Is one of the first routing models in FANET. In this model, an UAV loads data from a ground node; then the data is being carried to the destination by flying; and at the end the data reached to the destination ground node. The main objectives of **LCAD** routing are to maximize throughput and increase the security. Because of the use of a single UAV, data transmission delays are longer in **LCAD** but it achieves higher throughput. To solve this problem multi-UAVs system can be developed so that it decreases transmission delay as well as the distance among UAVs .It is possible to use this solution for delay tolerant and latency in sensitive bulk data transfer..

MLHR (MULTI LEVEL HIERARCHICAL ROUTING):

Model solves the scalability problems faced in large-scale FANET networks. By organizing UAV networks hierarchically a number of clusters needs to operate in different mission areas. Each cluster has a cluster head (**CH**), which will represent the whole cluster; this separate cluster can perform different activities. Each **CH** is in connection with the upper/lower layers (GCS, UAVs, satellites, etc.) directly or indirectly. To broadcast data and control info to other UAVs in the cluster, the **CH** should be in direct communication range of other UAVs in the cluster. This model is suitable if UAVs are controlled in changed swarms, the mission area is huge, and several UAVs are used in the network.

DCR(DATA CENTRIC ROUTING):

DCR is another static routing proposal. In fact, UAVs wireless communication supports one to many data transmission which is similar to one-to-one data transmission. This method is selected when the data is requested by a number of nodes, and data distribution is done by on-demand algorithms. **DCR** is a favorable model of routing mechanism and can be adjusted for FANET. This model can be chosen when the system contains a small number of UAVs on a planned path, which involves minimum assistance. It works well with cluster topologies where **CH** is responsible for disseminating information to other nodes in the cluster.

PROACTIVE ROUTING PROTOCOLS

Also known as table driven protocols, they use a table to store all routing information about all nodes in the network before transmitting data packets. The tables need to be refreshed when the topology changes. The main advantage of this type of routing is that it contains the updated information of the routes, thus, it is simple to select a path from the transmitter to the receiver. But, there are some explicit weaknesses. Firstly, due to the number of message exchanged between nodes, routing protocols cannot efficiently use bandwidth, which is a limited resource in FANET networks. Secondly, it acts slowly when the topology changes or when a failure occurs. Therefore as depicted in , this kind of protocols is not suitable for highly mobile and larger networks.

Despite this, several extensions of the **OLSR (Optimized Link State Routing Protocol)** MANET proactive protocol have been proposed for FANET.), **Directional OLSR (DOLSR)** is introduced in , for UAVs that were equipped with directional antennas. To further reduce communication overhead, a new forwarding mechanism was suggested in **DOLSR**.

Otherwise, **OLSR** is used in case of available omnidirectional antenna. If there is no available omnidirectional antenna, the UAV will go back to **DOLSR**. Compared to **OLSR**, **DOLSR** uses less Multi Point Relay (**MPR**) to achieve better performance in terms of lower communication overhead and shorter end-to-end delay. In , was proposed a speed-aware **Predictive-OLSR (POLSR)** protocol, where GPS information was exploited to aid the routing protocol. Based on GPS information, the relative speed between two nodes can be obtained, which is used to evaluate the communication link quality. Therefore, the UAVs with higher link quality are chosen for routing. In was proposed a Mobility and Load aware **OLSR (ML-OLSR)** protocol, also based on **OLSR**. Similar to **POLSR**, relative speed and position between neighbor nodes were considered in **ML-OLSR** to avoid selecting a high-speed node as **MPR**.

ON DEMAND ROUTING PROTOCOLS (REACTIVE ROUTING PROTOCOLS):

In this type of protocol, a route is established only when a node wants to join other nodes (i.e., for sending a packet) . This protocol can solve the overhead issue in proactive routing protocol. Two type of messages are exchanged which are Route Request and Route Reply. Route Request message is sent from the transmitter node to all the neighbors nodes using flooding techniques to scan the path, and each node uses the same method until it reaches the receiver. The second one Route Reply is created and sent by the destination node to the source node using the unicast method. In this situation, each node saves the currently used path, not all the paths, hence, there is no need to update all tables in the network. Many reactive routing protocols have been proposed explicitly for FANET, like Time slotted AODV and RGR (Reactive-Greedy-Reactive) that are both based on AODV (Ad-hoc On demand Distance Vector). Time slotted AODV is proposed in for FANETs, basically it is a time slotted version of AODV. Time slotted AODV uses dedicated time slots in which only one node can send data packet. Although it increases the use of network bandwidth but mitigates the packet collisions and ensure packet delivery.

HYBRID ROUTING PROTOCOLS

Hybrid routing is a combination of previous protocols (proactive and reactive), and is proposed to overcome their weakness. By using this protocol, the overhead of control messages in proactive protocols can be reduced and the important latency of the initial route discovery process in reactive protocols can be decreased. This type of protocols, is especially adaptable for large networks: the network is decomposed into a number of zones where routing in intra zone is done with the proactive method while inter zone routing is performed using the reactive method. In FANET, the mobility of node and link behavior makes it hard to obtain and maintain information. Hence, this makes adjusting routing strategies difficult to do. Example of hybrid routing protocols: TORA (Temporarily Ordered Routing Algorithm) mainly uses a reactive routing protocol but it also use some proactive protocol. It constructs and preserves a DAG (Directed Acyclic Graph) from the source node to the destination. TORA does not use a shortest path solution, some times longer routes are used to reduce network overhead. Each node has a parameter value termed as height in DAG, which is unique for each node. Data flow as a fluid from the higher nodes to lower ones. It is structurally loop free because data cannot flow to the node that has higher value. ZRP (Zone Routing Protocol) is based on the concept of zones. In this protocol, each node has a different zone. The zone is defined as the set of nodes whose minimum distance is predefined radius.

GEOGRAPHIC ROUTING PROTOCOLS

This category assumes pre-knowledge of geographic location of nodes. It uses node locations to find the best path from a source to a destination. Usually, there are two different strategies: greedy forwarding and a backup method in case the former fails. One of the earliest geographic routing protocols proposed for wireless communication is **GPSR (Greedy Perimeter Stateless Routing for wireless)**. In **GPSR**, each node tracks its neighbors locations through periodic beacons. Collected locations will be used to greedily forward data packets to the neighbor that is geographically closest to the final destination. Based on **GPSR** principal, many approaches have been lately proposed for inter UAV communications. Such as **MPGR (Mobility Prediction Geographic Routing)** which takes routing decisions based on geographic positions of UAVs. **MPGR** enhances **GPSR** by adding a mobility prediction method that reduces the impact of the high mobility of UAVs with an acceptable communication overhead. **USMP(UAV Search Mission Protocol)** is another approach based on the GPSR routing protocol. To improve the routing process, USMP adds two cross-layer features: the Location Up date that helps each UAV to determine the location that has not been scanned and the way point conflict resolution mechanism to avoid UAV collision. **GPMOR(Geographic Position Mobility Oriented Routing)** is another example of geographic protocol dedicated for FANET. Like MPGR, GPMOR introduces a mobility prediction module that allows each UAV to predict the movement of its neighboring nodes**. LAROD(Location Aware Routing for Opportunistic Delay tolerant network)** combines greedy forwarding and store-carry-and-forward techniques to accommodate to variable topology and UAVs mobility patterns. By using these routing protocols, a FANET can dynamically discover new paths between the source node and the destination node, and this network may permit the dynamic addition and subtraction of UAV nodes.

1.6 MOBILITY MODELS:[2]

Mobility models are used to simulate the movement of real mobile nodes (UAVs) and to mimic the realistic FANET environment. They represent the movement of UAVs node and how their position, acceleration and velocity vary over time. These models are used to verify whether a given protocol (for example routing protocol) is useful in a particular situation.

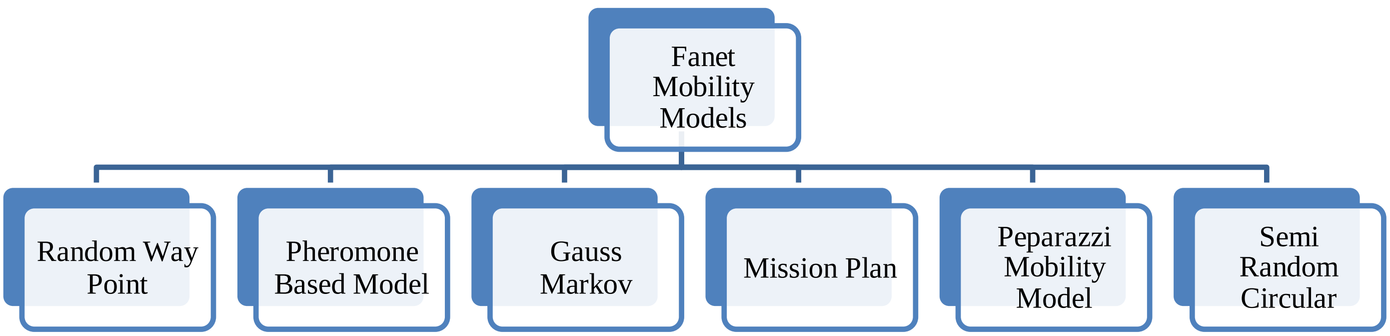

Fig 1.4: MOBILITY MODEL IN FANETS.

RANDOM WAY POINT (RWP):

mobility model is based on three actions, turn right, turn left and going straight . Each nodes choose randomly the destination moves with a random speed and pause time. When the pause time expire UAVs select another random location and moves with another random speed value. The flying nodes decide on their movements according to fixed probabilities. The RWP is the most mobility model used in the simulation scenarios. But unfortunately this model cannot be suitable for FANET networks, mainly, because UAVs do not change their speed and direction rapidly at one time and cannot stay for a moment at the same position.

PHEROMONE BASED MODEL (PBM):

Model is proposed based on the pheromone map. It takes into consideration the area specified for each flying nodes and the pheromones guide node developments. Each node marks the area that it checks on the map, and shares it by broadcasting toot her nodes. The pheromone map is used to guide nodes. The UAVs exchange information about their scanned area, and according to what they decide, they turn right, left or go straight. In order to expand the coverage, the UAVs nodes prefer the movement through the area with low pheromone smell.

GAUSS MARKOV (GM):

Mobility model uses one tuning parameter to change the degree of randomness. Initially, each UAV is set to a specific direction and speed. After that at each period of time, the action will refresh the direction and the speed of the nodes. The direction and speed are given based on the last position.

1.7 FANETS APPLICATIONS:

FANET is used in many domains , their applications cover different areas form agriculture , military ,and education

Fig 1.5: FANET Application Domains

1.8 Conclusion:

We tried through this chapter to focus on what are Flying Ad-Hoc Networks, their characteristics and their routing protocol, mobility models and communications topologies.

Chapter 02 : State Of The Art

2.1 INTRODUCTION:

The story of sensor networks can be tracked back to the 1960s, when the United States navy had deployed a sound surveillance system (**SOSUS**) using hydrophones (microphones deployed underwater) on the bottom of the ocean to detect submarines. They applied 40 hydrophones to trail, then, they extended the project to the entire east and west coasts. At that time, the transmissions were done through multi-conductor armored cables. Nowadays, wireless sensor nodes are used to detect the earthquakes in the pacific [4]. Indeed, a new class of networks has appeared in the last decade: the so-called wireless sensor network (WSN). They consist of individual nodes that are able to interact with their environment by sensing or controlling physical parameters; these nodes have to collaborate to fulfill their tasks as, usually, a single node is incapable of doing so; and they use wireless communication to enable this collaboration [5]. Wireless sensor networks are widely used in several applications such as military, environmental, health-care, and home applications.

2.2 WIRELESS SENSOR NETWORKS:

A wireless sensor network (**WSN**) is a network composed of a large number of low-power devices that sense the environment and send their readings to one or more sinks [6]. The devices composing a WSN are called sensor nodes, or motes, and they have the following characteristic

- they are small ;
- they have limited memory, processing power, and energy (most of them are battery powered) ;
- They are composed of sensing, data processing, and communication components.

- They have limited communication range and data-rate.

Generally, the WSNS are densely deployed because of these characteristics.

2.3 CATEGORIES OF SENSORS:

Handling a wide range of application types will hardly be possible with any single type of a sensor node. Nonetheless, certain common traits appear, especially with respect to the characteristics and the required mechanisms of such networks. Indeed, in the majority of applications, the sensors require readiness for field deployment in terms of economic and engineering efficiency. The scalability of the sensor is also important in distributed environmental monitoring tasks, which require that the sensors be small and inexpensive enough to scale up to many distributed systems. Sensors are deployed in hundreds of thousands. Therefore, it is expected that the cost will drop but current generation sensors are still expensive to allow widely deployment [7].

Sensors can be classified in terms of where they are deployed or used

SENSOR CATEGOR	PARAMETER	FIELD-READINE	SCALABILIT
PHYSICAI	TEMPERATURE	HIGH	HIGH
	HUMIDITY (SOIL, LEAF, AMBIE	HIGH	HIGH
	WIND (SPEED AND DIRECTIOI	HIGH	HIGH
	PRESSURE	HIGH	HIGH
CHEMICA	DISSOLVED OXYGEN	HIGH	HIGH
	PH	HIGH	HIGH
	HEAVY METALS	LOW	LOW

	NUTRIENTS (NITRATE, AMMONI	LOW-MEDIUM	LOW-HIGH
BIOLOGIC	MICROORGANISMS	LOW	LOW
	BIOLOGICALLY ACTIVE CONTAMINANTS	LOW	LOW

TABLE 2.1: SUMMARY OF SENSORS CATEGORIES [7]

2.4 HETEROGENEOUS WSN:

Sensor nodes can be heterogeneous by constructions, that is, some nodes have larger batteries, farther-reaching communication devices, or more processing power. they can also be heterogeneous by evolution, that is, all nodes started from an equal state, but because some nodes had to perform more tasks during the operation of the network, they have depleted their energy resources or other nodes had better opportunities to scavenge energy from the environment (e.g. nodes in shade are at a disadvantage when solar cells are used).

Whether by construction or by evolution, heterogeneity in the network is both a burden and an opportunity. Heterogeneous WSN consists of sensor nodes with different abilities, such as various sensor types and communication range, thus provides more flexibility in deployment. For example, we can construct a WSN in which nodes are equipped with different kinds of sensors to provide various sensing services [5].

Several issues are still to be solved in heterogeneous wireless networks such as determining the theoretical capacity of heterogeneous WSN, interpretability of different technologies, mobility, quality of service, and so on. There are several benefits to a heterogeneous WSN as opposed to a traditional homogeneous wireless network including increased reliability, improved spectrum efficiency, and increased coverage. Reliability is improved because when one particular access technology within the heterogeneous WSN fails, it may still be possible to maintain a connection by falling back to another access technology. Spectrum efficiency is improved by making use of access technologies which may have few users through the use of load balancing across access technologies and coverage may be improved because different access technologies may fill holes in coverage that any one of the single networks alone would not be able to fill.

2.5 DATA COLLECTION APPLICATIONS:

As aforementioned, WSNs are widely used in several applications ranging from military, agriculture to health monitoring. Data collection is one of the most important issues in WSN, and this problem has witnessed a significant amount of researches over the decades. Traditionally data gathering schemes based on the static topology where the nodes are statically deployed but later on mobile nodes which are more energy efficient. The main idea of data-gathering is to sense data and forward these data to the collectors for further processing. Traditionally, the data transmissions were done in multiple hops. The relay nodes help the simple nodes to forward their data. Thus, they die out (lose energy) Fastly which eventually lead to loss of coverage.

2.6 CLASSIFICATION OF THE DATA COLLECTION UNDERLYING PROTOCOLS

– ONE-HOP COMMUNICATION PROTOCOLS:

in this case, source nodes that are within the range of the destination node can directly communicate with the collectors. The transmission in this context requires appropriate distance between the source nodes and the destination nodes, and there are no obstacles between them and obstruct them to communicate with each other. If there are multiple nodes within the range of the destination node at the same time, the medium can be shared according to different protocols that can be further classified as contention-based protocols, contention-free protocols and hybrid protocols. Many protocols have been proposed on these three classifications, such as b-mac [8] (contention-based), flama [9] (contention-free), and z-mac [10] (hybrid).

– MULTI-HOP COMMUNICATION PROTOCOLS:

it is better for the source nodes that are out of the range of the destination node or have poor transmission conditions (e.g. high energy consumption, high packet losses, etc.) to send packets by means of intermediate nodes that are within the range of the destination node and the source node at the same time or have better communication situations. the group of intermediate nodes is usually called potential forwarders. The use of mechanisms mainly depends on the network information provided by each node.

GENERAL CLASSIFICATIONS OF ROUTING PROTOCOLS:

Several protocols are used in FANET. Some are more effective where others are simpler to implement. Most of routing protocols can be classified as data-centric, hierarchical, location-based or QOS-aware. DATA-centric protocols are query-based and depend on the naming of desired data. Hierarchical protocols aim at clustering the nodes so that cluster heads can do some aggregation and reduction of data to save energy. LOCATION-based protocols utilize the position information to relay the data to the desired regions. The QOS-aware are based on general network flow modeling for meeting some **QOS** requirements.

– DATA-CENTRIC PROTOCOLS:

Since transmitting data from each sensor node within the deployment region might result in significant unnecessary redundancy in data and incur in unnecessary energy and traffic expenditure, routing protocols that can select a set of sensor nodes and utilize data aggregation during the relaying of data have been considered.

in data-centric routing, the sink sends queries to selected regions which might be selected using clusters and waits for data from the nodes located in the regions. Since data is being requested through queries, attribute-based naming is necessary to specify the properties of data. Sensor protocols for information via negotiation (spin) [11] is the first data-centric protocol, which considers data negotiation between nodes to eliminate redundant data and save energy. Later, directed diffusion [12] has been developed and has become a breakthrough in data-centric routing. Many other protocols have also been proposed based on directed diffusion, such as energy-aware routing [13], rumor routing [13] and gradient-based routing [15].

– HIERARCHICAL ROUTING PROTOCOLS:

Hierarchical clustering, originally proposed in wired networks, is well-known technique in WSNs. The hierarchical routing protocol is an energy efficient approach through sensor nodes, base station, and cluster heads. The main aim of cluster-based routing is to efficiently maintain the energy consumption of sensor nodes and improve network lifetime. In a hierarchical architecture, higher-energy nodes can be used to process and send the information, while low-energy nodes can be used to sense the target. The introduction of clusters can greatly contribute to overall system scalability, lifetime, and energy efficiency. CLUSTER-based routing is mainly two-layer routing where one layer is used to select cluster heads and the other for routing. Cluster formation is typically based on the energy reserve of sensors and sensor's proximity to the cluster head. Low energy adaptive clustering hierarchy (leach) [16] is an initial hierarchical routing approach which considers homogenous wireless sensor network. The selection of cluster heads in leach depends on the highest residual energy. LEACH rotates cluster head to evenly distribute the energy load among the sensors in the network and extend the network lifetime.

– LOCATION-BASED PROTOCOLS:

most of the routing protocols for sensor networks require location information of sensor nodes. in most cases, location information is needed to calculate the distance between two particular nodes so that energy consumption can be estimated. Relative coordinates of neighboring nodes can be obtained by exchanging such information between neighbors [17]. Alternatively, the location of the nodes may be directly available using GPS if nodes are equipped with a small low-power GPS receiver. LOCATION-aided routing (LAR) [18] is a location-based routing protocol with an objective to limit the area to build a new route to a smaller request zone. in LAR, the route requests were sent to the whole network. senouci et al. [19] optimized LAR, and only the nodes in the request zone have opportunities to forward the requests. thus, the routing overhead is widely reduced.

– QOS-AWARE PROTOCOLS:

The network needs to ensure quality of service (QOS) besides ease of implement, energy efficiency and low cost. ONE of the major design goals of WSNs is reliable data communication under minimum energy depletion to extend the lifetime of the network. Some of the routing challenges and design issues that affect the routing process in WSNs are : node deployment, coverage, connectivity, node and link heterogeneity, fault tolerance, scalability, transmission media, data aggregation, and QOS. In QOS-based routing protocols, the network has to balance between energy consumption and data quality. Particularly, the network has to satisfy certain QOS metrics (delay, energy, bandwidth) when delivering data to the base station. Speed [20] is a QOS-aware routing protocol which is designed for real-time communication in sensor networks. Speed handles congestion and provides soft real-time communication by using feedback control and non-deterministic geographic forwarding. a node computes speed to each neighbor and then forwards the packet to a neighbor which is close to the destination and has higher speed than other neighbors. in [21], marot et al. propose a local load balancing routing protocol which aims to help source nodes to apply neighbor nodes potential capabilities without knowing their information. Their protocol improves the reliability and efficiency of the link quality indicator.

MULTI-HOP PROTOCOLS:

Several multi-hop protocols exist in the literatures, they can be categorized as deterministic and random, according to the topologies.

(A) DETERMINISTIC TOPOLOGIES:

– STATIC ROUTING

This scheme establishes a pre-computed static table that is loaded when initializing the network. Unlike a dynamic routing protocol, static routes are not automatically updated and must be manually reconfigured by a network administrator when the network topology changes. Static routing provides ease of routing table maintenance in small scale networks. The route used to send data is known in advance. thus, it uses little bandwidth as routers do not exchange routes. However, configuring the route table is time-consuming and error-prone, especially in large scale networks.

Load-carry-and-deliver (LCAD) [22] is a static routing protocol using UAVs to relay messages between two ground nodes. The route is configured on the ground before takeoff. in LCAD, the authors object to maximize throughput by configuring nodes positions. This type routing protocols are used for repetitive tasks, such as periodically surveillance missions. Adaptive routing using clustered hierarchies (arch) [23] creates a multi-level hierarchy that adjusts its depth dynamically in response to the network topology. These hierarchical routing protocols, such as arch which is based on multilevel clustering, consist of a number of different components, such as clustering, routing and location management. Here, clustering is the process by which nearby nodes form groups, called clusters. In [24], the authors study the theoretical scalability aspects of multi-level hierarchical routing in MANET. in the general scheme they analyze, nodes organized in clusters, which are then grouped in higher level clusters. The number of levels is logarithmic in the network size. [24] is one of the crucial papers with comprehensive theoretical results of multi-level hierarchical routing protocols.

– DYNAMIC ROUTING:

Unlike static routing, dynamic routing helps the network administrator to manage the time-consuming and exacting process of configuring and maintaining static routes. Dynamic routing is able to find remote networks, maintain the routing information, and select the best path to destinations. Therefore, it is suitable in all topologies when multiple routers are required.

1. APPLICATION IN CONNEX TOPOLOGIES:

The typical applications in connex topology are based on mobile ad hoc networks (MANET). They can be divided into proactive, reactive and hybrid routing protocols.

– PROACTIVE ROUTING PROTOCOLS:

In WSNs, sensors are used to store routing information for a specific region of the network. However, many of the tables must be updated when the topology is changed. Proactive routing protocols are based on periodic exchange of control messages. The main advantage of proactive routing is that it immediately provides the required routes when needed. Thus, a large number of messages are required to keep the system up to date. However, negative points are also present such as the bandwidth constraints on each communication link or the addition of a delay to each topological change (slow reaction). this will present a real disadvantage for time sensitive applications.

Optimized link state routing (OLSR [25]) is a proactive routing protocol used for mobile sensor networks. it maintains the topology information of the network through exchanging messages periodically at each node.

Furthermore, multi-point relaying scheme is used to efficiently and economically flood its control messages. OLSR provides optimal routes regarding the number of hops, which are immediately available when needed. OLSR is an optimization protocol over a pure link state. Other examples of this kind of protocols are DSDV [26], star [27] and TBRPF [28].

– REACTIVE ROUTING PROTOCOLS:

In reactive routing approach, a routing protocol does not respond to finding a route to a destination node, until it has a reservation request. The reactive routing protocol attempts to find a route only on-required by flooding its query in the sensor networks. The routing information will be stored only for the duration of the communication.

The removal of exchange messages periodicity improves the availability of bandwidth and eliminates the control traffic overhead. However, the implementation and closure of roads occurs more frequently.

It also increases the latency in finding a route to a destination node.

There are two classes of reactive protocols : source routing and hop-by-hop routing. In source routing, each packet contains the full path in its header, so the nodes only make switching, and no periodic information is required to maintain the connectivity. Dynamic source routing (DSR) [59] protocol is an example of source routing protocols. it adapts quickly to routing changes when host movement is frequent, yet requires little or no overhead during periods in which hosts move less frequently. However, source routing has bad scalability because one of the sensitive points remains the loss of a communication link, and the larger the network size, the greater the likelihood of loss.

in the case of hop-by-hop routing, each node stores and maintains routing information to be able to switch packets. All the nodes must also be attentive to their neighbors, and to the periodic exchange of messages. ad-hoc on demand distance vector routing (AODV) [29]) is a hop-by-hop reactive routing protocol. in AODV, each mobile host operates as a specialized router, and routes are obtained as needed (i.e., on-demand) with little or no reliance on periodic requirements. it is suitable for a dynamic self-starting network, as required by users wishing to utilize ad-hoc networks. Based on AODV, senouci et al. [30] use energy consumption as a routing metric and propose three extensions (LEAR-AODV, par-AODV, and LPR-AODV) to the shortest-path routing algorithm. The three algorithms reduce the nodes energy consumption through routing packets using energy-optimal routes. labiod et al. [31] present a comprehensive performance comparison of five multipath reactive routing in ad hoc networks : three node-disjoint multipath routing protocols and two routing protocols based on a untrusted node disjoint path scheme.

– HYBRID ROUTING PROTOCOLS:

The hybrid routing protocols bring together the advantages of the two techniques to reduce the impact of the weaknesses in each type. They keep routes available for some destinations all the time, but find routes for other destinations when required. Hybrid protocols minimize the latency of reactive protocols and reduce the overhead of proactive protocols.

Hybrid routing will be particularly efficient when the network is divided into several zones where inter-zone and intra-zone routing is applied (inter-zone with reactive routing and intra-zone with proactive routing). [32] Shows an analysis of zone routing protocol (ZRP) in MANET.

However, this strategy remains difficult to implement due to the dynamics of the nodes and their continuous activities.

2. APPLICATION IN DELAY TOLERANT NETWORKS:

The primary focus of researchers studying on delay tolerant networks (DTNS) are routing issues. Many studies have been performed on how to handle the sporadic connectivity between the nodes and provide a successful and efficient delivery of messages to the destination. one of the pioneering algorithms for DTNS is epidemic routing [33] which is published by Vahdat and Becker. the source nodes in the epidemic scheme can send packets to all neighborhood nodes without any filter. Thus, the source nodes may build surplus routes which may never be used in communication.

in some large-scale networks, it is complex to take into account the details of each node, the global information of the network topology would be a better choice. The characteristics of the wireless links and nodes are the main factors when selecting and prioritizing the potential forwarders. In other words, the calculated metric for each node depends on the cost of the remaining path from the neighbors of a source node to its destination node.

The opportunistic routing in dynamic ad hoc networks (OPRAH) [34] is a simple hops-count mechanism reflecting the number of hops that build the route between two nodes. In OPRAH, the nodes with a smaller number of hops to the destination node are a better choice than those with a larger amount of hops. Each node needs to know the topology information and its hops-count to the destination. The OPRAH metric does not consider the delivery ratio between the source nodes and the PFS. Thus, in OPRAH, the source node may select some of its neighbors with smaller hops-counts, while the transmission rate to reach them is very low. Another metric that is also based on traditional routing protocols is the expected transmission count metric (ETX) [35]. ETX measures the average number of times that a packet must be transmitted or retransmitted on a link or on a route, to be received by the destination node. ETX serves to calculate and implement if knew the link delivery probability between the source nodes and the PFS. However, authors in [36] presented that the ETX does not always find an acceptable potential relay node. lu et al. [37] showed that the performance of the network might degrade when the combination of or with ETX is applied.

(B)RANDOM TOPOLOGIES:

In some applications, such as animal tracking or rescuing, the targets move randomly, and consequently the network topology is randomly changing. Random routing protocols play a crucial role in such context.

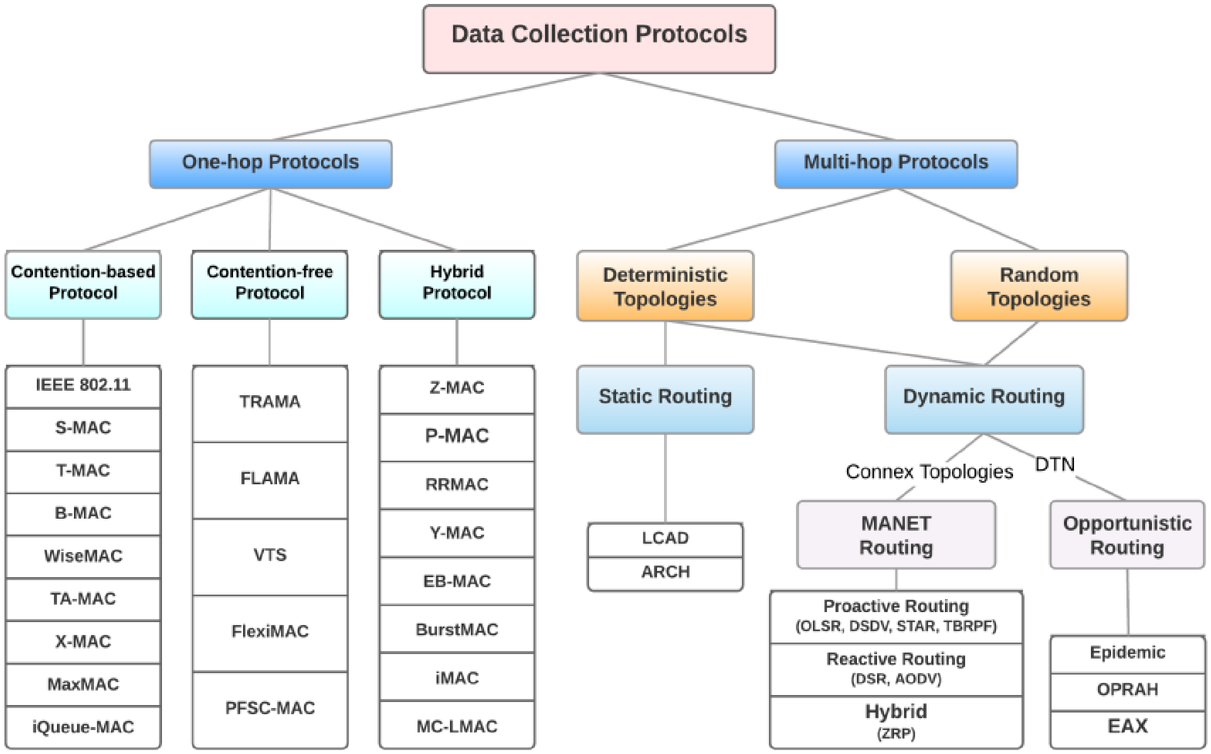

FIG1.2: THE CATEGORIZES OF DATA COLLECTION UNDERLYING PROTOCOLS

2.7 DISCUSSION:

In practical applications, we can conclude which protocol is better for this application according to the criteria the application needs. for example, if we apply UAV-assisted WSNs to realize video streaming which needs high data throughput, low energy and latency. The standard, IEEE 802.11 works well on data throughput, efficiency energy and low latency. Moreover, on other criteria, such as communication range, fairness and traffic adaptability also suitable to this application. Thus, we can choose IEEE 802.11 in this application. Additionally, there are many other conditions we needs to consider when we choose a mac protocol.

CHAPTER 03: Implementing Leach Protocol For UAV-Assisted WSN In OMNET++ Environment.

3.1 Introduction:

Collection data from wireless sensor camera (Streaming Videos) consumes big amounts of energy, Thus using standard Leach (Low Energy Adaptive Clustering Hierarchical protocol) alone do not makes big different . Therefor We propose to use flying nodes to ameliorate Leach to minimize transmission and receiving energy consumed by ground nodes.

3.2 OMNET++ Simulator[40]:

OMNET++ is an extensible, modular, component-based C++ simulation library and framework, primarily for building network simulators. "Network" is meant in a broader sense that includes wired and wireless communication networks, on-chip networks, queuing networks, and so on. Domain-specific functionality such as support for sensor networks, wireless ad-hoc networks, Internet protocols, performance modeling, photonic networks, etc., is provided by model frameworks, developed as independent projects. OMNeT++ offers an Eclipse-based IDE, a graphical runtime environment, and a host of other tools. There are extensions for real-time simulation, network emulation, database integration, SystemC integration, and several other functions. OMNeT++ is distributed under the Academic Public License.

Although OMNeT++ is not a network simulator itself, it has gained widespread popularity as a network simulation platform in the scientific community as well as in industrial settings, and building up a large user community.

OMNeT++ provides a component architecture for models. Components *(modules)* are programmed in C++, Then assembled into larger components and models using a high-level language *(NED)*. Reusability of models comes for free. OMNeT++ has extensive GUI support, and due to its modular architecture, the simulation kernel (and models) can be embedded easily into your applications.

Components:

The main ingredients of OMNeT++ are:

- Simulation kernel library (C++)
- The NED topology description language
- Simulation IDE based on the Eclipse platform
- Interactive simulation runtime GUI (Qtenv)
- Command-line interface for simulation execution (Cmdenv)
- Utilities (makefile creation tool, etc.)
- Documentation, sample simulations, etc.

Models

During the years OMNeT++ has been available, countless simulation models and model frameworks have been written for it by researchers in diverse areas: queuing, resource modeling, internet protocols, wireless networks, switched LANs, peer-to-peer networks, media streaming, mobile ad-hoc networks, mesh networks, wireless sensor networks, vehicular networks, NoCs, optical networks, HPC systems, cloud computing, SANs, and more. Most of these model frameworks are open source, developed as independent projects, and follow their own release cycles.

Platforms

The OMNeT++ simulation kernel is standard C++, and runs basically on all platforms where a modern C++ compiler is available. The Simulation IDE requires Windows, Linux, or macOS.

3.3 INET FRAMEWORK [39] :

INET Framework is an open-source model library for the OMNeT++ simulation environment. It provides protocols, agents and other models for researchers and students working with communication networks. INET is especially useful when designing and validating new protocols, or exploring new or exotic scenarios.

INET contains models for the Internet stack (TCP, UDP, IPv4, IPv6, OSPF, BGP, etc.), wired and wireless link layer protocols (Ethernet, PPP, IEEE 802.11, etc.), support for mobility, MANET protocols, DiffServ, MPLS with LDP and RSVP-TE signaling, several application models, and many other protocols and components.

Several other simulation frameworks take INET as a base, and extend it into specific directions , such as vehicular networks, overlay/peer-to-peer networks, or LTE.

INET Designed for Experimentation

INET is built around the concept of modules that communicate by message passing. Agents and network protocols are represented by components, which can be freely combined to form hosts, routers, switches, and other networking devices. New components can be programmed by the user, and existing components have been written so that they are easy to understand and modify.

INET benefits from the infrastructure provided by OMNeT++. Beyond making use of the services provided by the OMNeT++ simulation kernel and library (component model, parameterization, result recording, etc.), this also means that models may be developed, assembled, parameterized, run, and their results evaluted from the comfort of the OMNeT++ Simulation IDE, or from the command line.

Some features:

- OSI layers implemented (physical, link-layer, network, transport, application)
- Pluggable protocol implementations for various layers
- IPv4/IPv6 network stack (or build your own network layer)
- Transport layer protocols: TCP, UDP, SCTP
- Routing protocols (ad-hoc and wired)
- Wired/wireless interfaces (Ethernet, PPP, IEEE 802.11, etc.)

- Physical layer with scalable level of detail (unit disc radio to detailed propagation models, frame level to bit/symbol level representation, etc.)
- Wide range of application models
- Network emulation support
- Mobility support
- Supports the modeling of the physical environment (obstacles for radio propagation, etc.)
- Separation of concerns
- Visualization support

3.4 AVENS FRAMEWORK(A Novel Flaying Ad-Hoc Network Simulator)[38]:

AVENS is part of a major research project concerning the provision of a test bed to simulate UAV (Unmanned Aerial Vehicle) flight and control, using different, controlled and changeable configurations. The main intention of **AVENS** is to provide a simulation test bed for virtual experiments of network coverage and connectivity among UAVs flying in cooperation or sharing the same airspace.

The purpose of **AVENS** is to offer a platform for mobile ad hoc networks analysis where UAVs are mobile nodes sharing the wireless medium for exchanging messages. The goal is to use a flight simulator for controlling the aerial vehicles and a network simulator for obtaining network measurements such as transmission rate, good put, RSSI (Received Signal Strength Indication), throughput, package loss, number of retransmission etc.

3.5 SETTING UP THE ENVIRONMENT:

After importing the two projects INET and AVENS in OMNET IDE we need to link AVENS project with the INET framework.

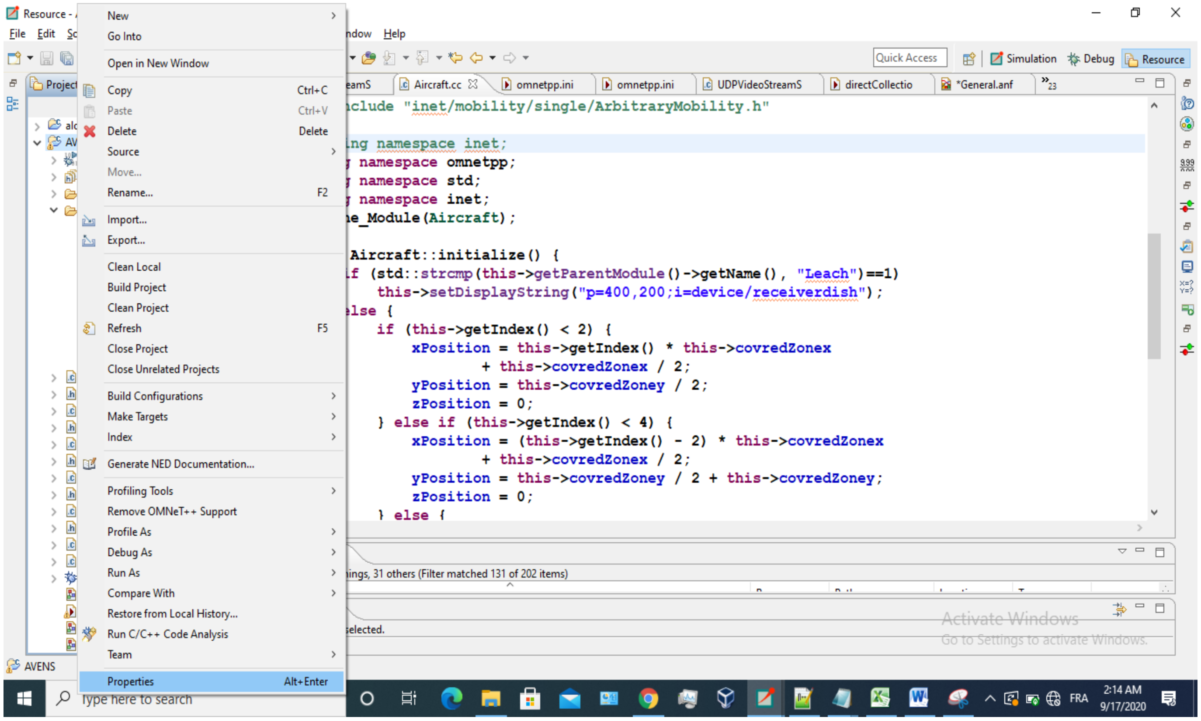

FIG 3.1.1: referencing AVENS project with INET framework

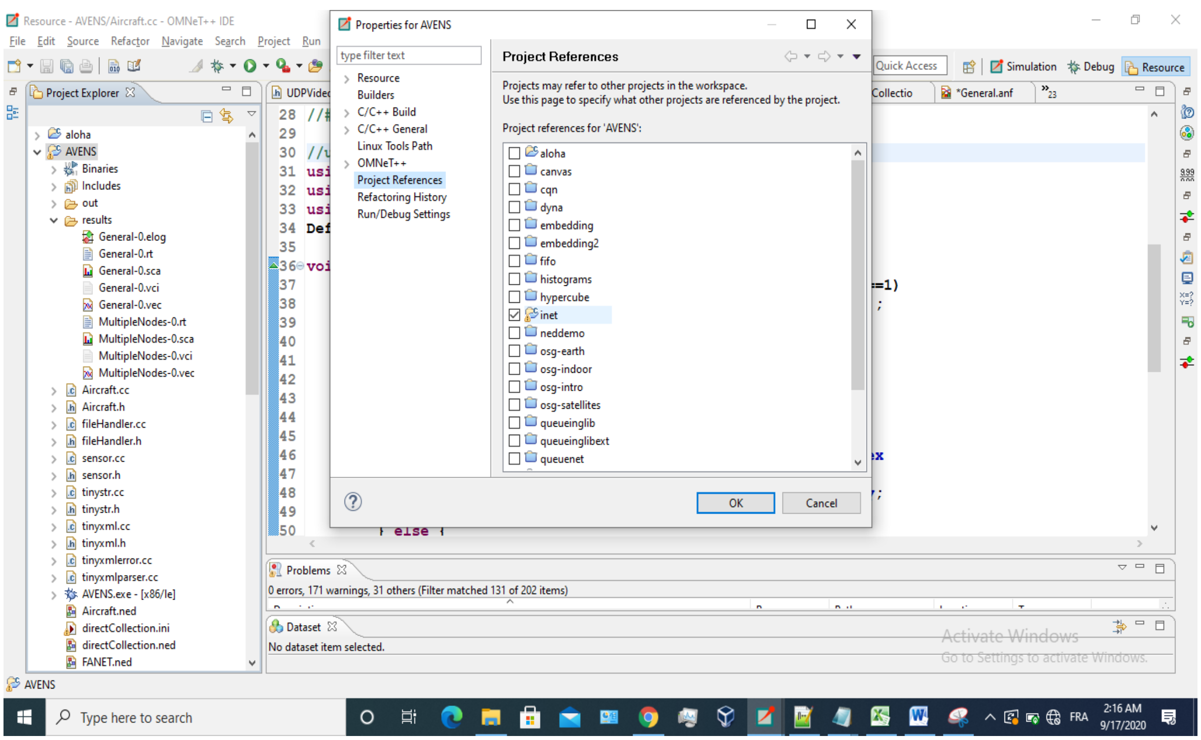

Fig 3.1.2: Referencing AVENS project with INET framework

3.6 IMPLEMENTING LEACH PROTOCOL:

In this study two approaches are proposed:

- **Direct Communication** :

  In this approach flying nodes (**FN**) collect directly their member's video in the ground

  After sending ClusterMessage with multicast Address ground nodes (**GN**) executes **ARP** (Address resolution protocol) protocol to resolve the IP address of **FN** associated to then stream their video.

- **Communication With Elected Cluster Head :**

  In this approach the **FN** collect their member's videos through cluster head.

Step1 ”election order ” : **FN** send ClusterMessage to the **GN.**

Step 02 “Clustering step” : after receiving **GN** ClusterMessage **GN** generates random number Based on this value the **GN** becomes cluster head **CH** or not.

Step 03 “Joining members ” : after the **GN** became **CH** sends request join message to **GN** in its communication range with multicast address .

Step 04 “Streaming video to CH” : GN execute Arp protocol to resolve cluster head address then stream the video to cluster head.

Step 05 “video aggregation ” : after receiving videos the CH aggregates the videos then send the result to **FN**.

Parameter	Value

Space	800x400
Node number	20
Communication technology	802.11

Table 3.1: simulation parameters.

Leach Scenario:

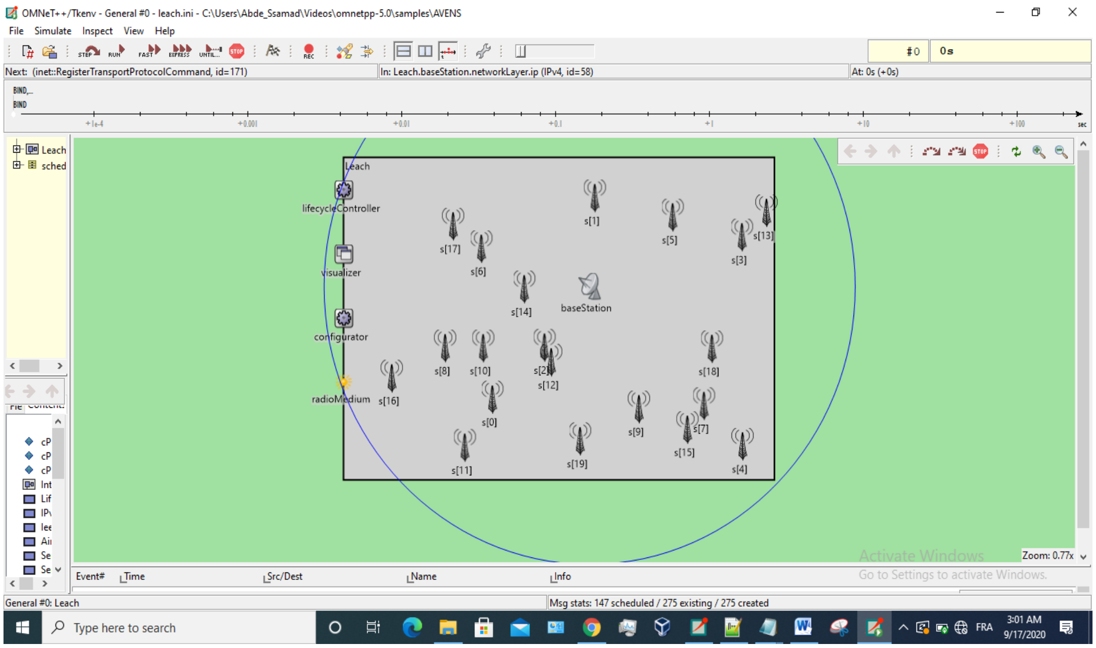

Fig3.2 : Leach Protocol Topology.

Table 3.1 : Leach Network description file

```
package avens;
import inet.common.lifecycle.LifecycleController;
import
inet.networklayer.configurator.ipv4.IPv4NetworkConfigurator;
import
inet.physicallayer.ieee80211.packetlevel.Ieee80211ScalarRadioM
um;
import inet.networklayer.ipv4.RoutingTableRecorder;
import inet.visualizer.integrated.IntegratedCanvasVisualizer;
network Leach
{
     parameters:
        int maxWidth = 800;
        int maxHeight = 400;
        @display("bgb=$maxWidth,$maxHeight");
        string networkType = "leach";
    submodules:
```

```
        visualizer: IntegratedCanvasVisualizer {
            parameters:
                @display("p=0,150");
        }
        lifecycleController: LifecycleController {
            @display("p=0,50");
        }
        configurator: IPv4NetworkConfigurator {
            parameters:
                @display("p=0,250");
                config = xml("<config><interface hosts='*'
address='192.168.0.x' netmask='255.255.255.0'/></config>");
        }
        radioMedium: Ieee80211ScalarRadioMedium {
            parameters:
                @display("p=0,350");
        }
        baseStation: AdhocHost {
            parameters:
                @display("p=450,250;i=device/receiverdish");
        }
        s[20]: Sensor {
            parameters:
                @display("p=250,10");
```

```
            }


}
```

Table 3.2: the base station class ”Client”

```
#ifndef __INET_UDPVIDEOSTREAMSVR_H

#define __INET_UDPVIDEOSTREAMSVR_H

#include <map>

#include "inet/common/INETDefs.h"

#include "inet/applications/base/ApplicationBase.h"

#include "inet/transportlayer/contract/udp/UDPSocket.h"

namespace inet {

class INET_API UDPVideoStreamSvr: public ApplicationBase {

public:

   virtual void processStreamRequest(cMessage *msg);

   virtual void sendStreamData(cMessage *timer);

   virtual void initialize(int stage) override;

   virtual void finish() override;

   virtual void handleMessageWhenUp(cMessage *msg) override;

   virtual void clearStreams();

   virtual bool handleNodeStart(IDoneCallback *doneCallback) override;

   virtual bool handleNodeShutdown(IDoneCallback *doneCallback) override;

   virtual void handleNodeCrash() override;

   void receiveStream(cPacket *msg);

   virtual bool getThreshold(float p);

   virtual std::string getMessageName(std::string msgName,int round);

   virtual int getRoundNumber(std::string msgName,std::string message);
```

```
virtual bool stringAtBegeningContains(std::string str1,std::string str2);



virtual void videoAggregation(cMessage* msg);

virtual void requestDataFromMembers(cMessage* msg);

virtual void sendDataToCH(cMessage* msg);
```

Table 3.3: leach Sensor class ”Server”

```
#include "inet/common/INETDefs.h"

#include "inet/applications/base/ApplicationBase.h"

#include "inet/transportlayer/contract/udp/UDPSocket.h"

#include <string>

namespace inet {

class INET_API UDPVideoStreamCli : public ApplicationBase

{

  protected:



    // state

    UDPSocket socket;

    cMessage *selfMsg = nullptr;



    cPar* scheduleTime=nullptr;

    // statistics
```

```
    static simsignal_t rcvdPkSignal;

    int round=1;

    protected:

    virtual int numInitStages() const override { return NUM_INIT_STAGES; }

    virtual void initialize(int stage) override;

    virtual void finish() override;

    virtual void handleMessageWhenUp(cMessage *msg) override;



    virtual void requestStream();

    virtual void receiveStream(cPacket *msg);



    // ApplicationBase:

    virtual bool handleNodeStart(IDoneCallback *doneCallback) override;

    virtual bool handleNodeShutdown(IDoneCallback *doneCallback) override;

    virtual void handleNodeCrash() override;

    virtual std::string getMessageName(int round);

    virtual int getRoundNumber(std::string msgName);

}
```

Direct Data Collection scenario:

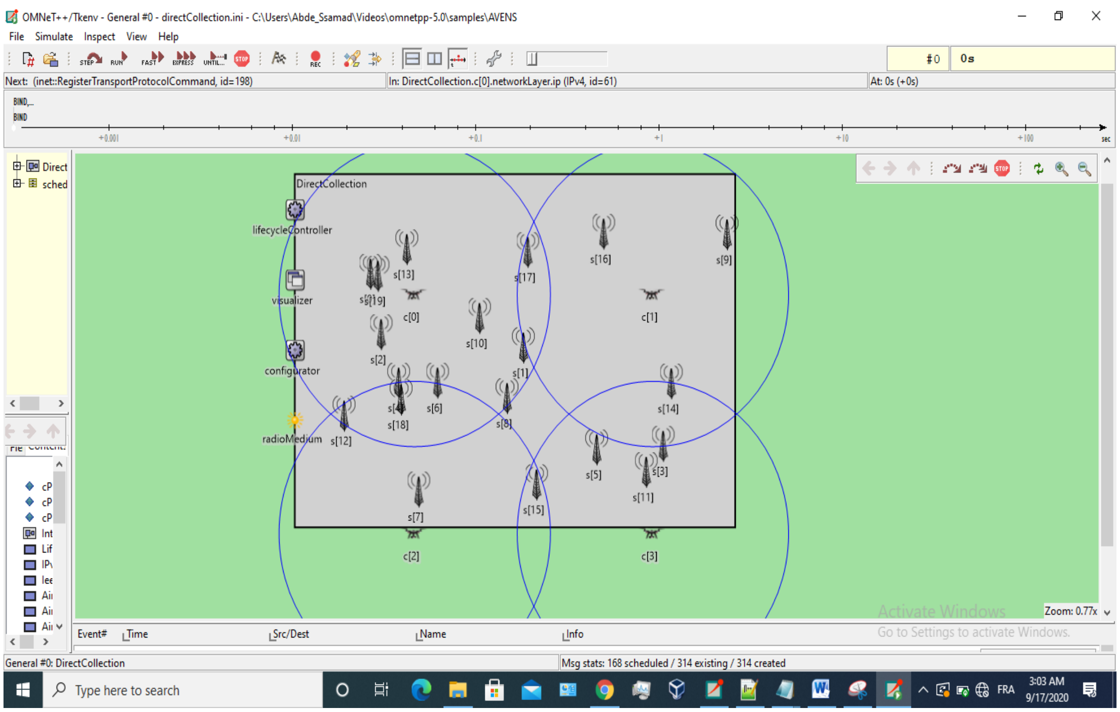

Fig3.3: Direct Data Collection Topology.

Table 3.4: Direct Data Collection Network Description File

```
package avens;

import inet.common.lifecycle.LifecycleController;

import

inet.networklayer.configurator.ipv4.IPv4NetworkConfigurator

import
```

```
inet.physicallayer.ieee80211.packetlevel.Ieee80211ScalarRad
edium;
import inet.networklayer.ipv4.RoutingTableRecorder;
import inet.visualizer.integrated.IntegratedCanvasVisualize
import inet.node.inet.NodeBase;


network DirectCollection
{
    parameters:
        int maxWidth = 800;
        int maxHeight = 400;
        @display("bgb=$maxWidth,$maxHeight");
        string networkType = "leach";


    submodules:
        visualizer: IntegratedCanvasVisualizer {
            parameters:
                @display("p=0,150");
        }
        lifecycleController: LifecycleController {
            @display("p=0,50");
        }
        configurator: IPv4NetworkConfigurator {
            parameters:
```

```
                @display("p=0,250");
                config = xml("<config><interface hosts='*'
address='192.168.0.x' netmask='255.255.255.0'/></config>");
        }
        radioMedium: Ieee80211ScalarRadioMedium {
            parameters:
                @display("p=0,350");
        }
        c[4]: Aircraft {
            parameters:
@display("p=400,200;i=device/receiverdish");
        }
        s[20]: Sensor {
            parameters:
                @display("p=250,10");
        }

}
```

Table 3.5: Direct Data collection Flying node Class

```
#ifndef __INET_UDPVIDEOSTREAMCLI_H

#define __INET_UDPVIDEOSTREAMCLI_H

#include "inet/common/INETDefs.h"

#include "inet/applications/base/ApplicationBase.h"

#include "inet/transportlayer/contract/udp/UDPSocket.h"

#include <string>

namespace inet {

class INET_API UDPVideoStreamCli : public ApplicationBase

{

  protected:

    // state

    UDPSocket socket;

    cMessage *selfMsg = nullptr;


    cPar* scheduleTime=nullptr;

    // statistics

    static simsignal_t rcvdPkSignal;

int round=1;

  protected:

    virtual int numInitStages() const override { return NUM_INIT_STAGES; }

    virtual void initialize(int stage) override;
```

```
virtual void finish() override;

virtual void handleMessageWhenUp(cMessage *msg) override;



virtual void requestStream();

virtual void receiveStream(cPacket *msg);



// ApplicationBase:

virtual bool handleNodeStart(IDoneCallback *doneCallback) override;

virtual bool handleNodeShutdown(IDoneCallback *doneCallback) override;

virtual void handleNodeCrash() override;

virtual std::string getMessageName(int round);

virtual int getRoundNumber(std::string msgName);}
```

Table 3.6: Direct Collection ground node class "Server"

```
#ifndef __INET_UDPVIDEOSTREAMSVR_H

#define __INET_UDPVIDEOSTREAMSVR_H

#include <map>

#include "inet/common/INETDefs.h"

#include "inet/applications/base/ApplicationBase.h"

#include "inet/transportlayer/contract/udp/UDPSocket.h"

namespace inet {

class INET_API UDPVideoStreamSvr: public ApplicationBase {
```

```
public:
   virtual void processStreamRequest(cMessage *msg);
   virtual void sendStreamData(cMessage *timer);
   virtual void initialize(int stage) override;
   virtual void finish() override;
   virtual void handleMessageWhenUp(cMessage *msg) override;
   virtual void clearStreams();
   virtual bool handleNodeStart(IDoneCallback *doneCallback) override;
   virtual bool handleNodeShutdown(IDoneCallback *doneCallback) override;
   virtual void handleNodeCrash() override;
   void receiveStream(cPacket *msg);
   virtual bool getThreshold(float p);
   virtual std::string getMessageName(std::string msgName,int round);
   virtual int getRoundNumber(std::string msgName,std::string message);
   virtual bool stringAtBegeningContains(std::string str1,std::string str2);


   virtual void videoAggregation(cMessage* msg);
   virtual void requestDataFromMembers(cMessage* msg);
   virtual void sendDataToUav(cMessage* msg);
}
```

Collecting data through cluster head Scenario:

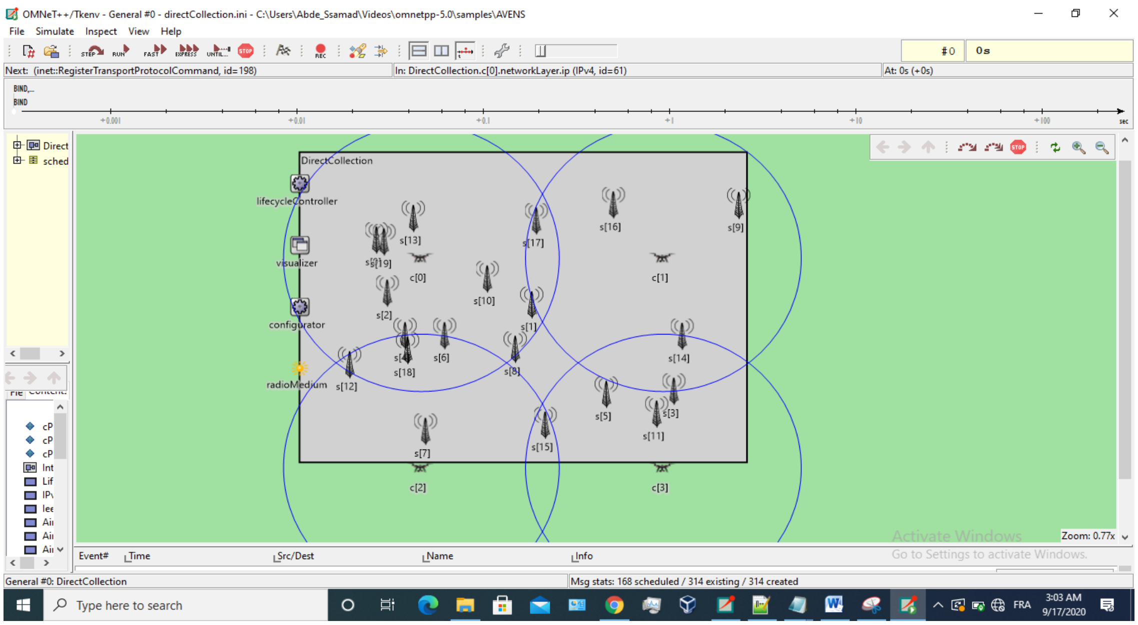

Figure 3.4: Leach with flying nodes Topology.

Table 3.7 : Leach with FN Network Description File

```
package avens;

import inet.common.lifecycle.LifecycleController;
import
inet.networklayer.configurator.ipv4.IPv4NetworkConfigurator
import inet.node.inet.AdhocHost;
import inet.node.inet.Router;
```

```
import
inet.physicallayer.ieee80211.packetlevel.Ieee80211ScalarRad
edium;
import inet.networklayer.ipv4.RoutingTableRecorder;
import inet.visualizer.integrated.IntegratedCanvasVisualize

import inet.node.inet.NodeBase;





network FANET
{
    parameters:

        @display("bgb=$maxWidth,$maxHeight");
        string networkType = "leach";

    submodules:
        visualizer: IntegratedCanvasVisualizer {
            parameters:
                @display("p=0,100");
        }
        lifecycleController: LifecycleController {
```

```
            @display("p=0,20");
        }



        configurator: IPv4NetworkConfigurator {
            parameters:
                @display("p=0,200");
                config = xml("<config><interface hosts='*'
address='192.168.0.x' netmask='255.255.255.0'/></config>");
        }
        radioMedium: Ieee80211ScalarRadioMedium {
            parameters:
                @display("p=0,0");
        }



        c[4]: Aircraft {
            parameters:
                @display("p=400,200;i=device/receiverdish")
        }
        s[20]: Sensor {
            parameters:
                @display("p=250,10");
        }
```

```
}
```

Table 3.8: Collection through cluster head Flying node Class “Client”

```
#ifndef __INET_UDPVIDEOSTREAMCLI_H

#define __INET_UDPVIDEOSTREAMCLI_H

#include "inet/common/INETDefs.h"

#include "inet/applications/base/ApplicationBase.h"

#include "inet/transportlayer/contract/udp/UDPSocket.h"

#include <string>

namespace inet {

class INET_API UDPVideoStreamCli : public ApplicationBase

{

  protected:

    // state

    UDPSocket socket;

    cMessage *selfMsg = nullptr;



    cPar* scheduleTime=nullptr;

    // statistics
```

```
  static simsignal_t rcvdPkSignal;

int round=1;

 protected:

  virtual int numInitStages() const override { return NUM_INIT_STAGES; }

  virtual void initialize(int stage) override;

  virtual void finish() override;

  virtual void handleMessageWhenUp(cMessage *msg) override;


  virtual void requestStream();

  virtual void receiveStream(cPacket *msg);


  // ApplicationBase:

  virtual bool handleNodeStart(IDoneCallback *doneCallback) override;

  virtual bool handleNodeShutdown(IDoneCallback *doneCallback) override;

  virtual void handleNodeCrash() override;

  virtual std::string getMessageName(int round);

  virtual int getRoundNumber(std::string msgName);

}
```

Table 3.9: Collection through cluster head ground node Class “Server”

```
#ifndef __INET_UDPVIDEOSTREAMSVR_H

#define __INET_UDPVIDEOSTREAMSVR_H

#include <map>
```

```
#include "inet/common/INETDefs.h"

#include "inet/applications/base/ApplicationBase.h"

#include "inet/transportlayer/contract/udp/UDPSocket.h"



namespace inet {

class INET_API UDPVideoStreamSvr: public ApplicationBase {

   virtual void processStreamRequest(cMessage *msg);

   virtual void sendStreamData(cMessage *timer);

   virtual void initialize(int stage) override;

   virtual void finish() override;

   virtual void handleMessageWhenUp(cMessage *msg) override;



   virtual void clearStreams();



   virtual bool handleNodeStart(IDoneCallback *doneCallback) override;

   virtual bool handleNodeShutdown(IDoneCallback *doneCallback) override;

   virtual void handleNodeCrash() override;

   void receiveStream(cPacket *msg);

   virtual bool getThreshold(float p);

   virtual std::string getMessageName(std::string msgName,int round);

   virtual int getRoundNumber(std::string msgName,std::string message);

   virtual bool stringAtBegeningContains(std::string str1,std::string str2);
```

```
virtual void videoAggregation(cMessage* msg);

virtual void requestDataFromMembers(cMessage* msg);

virtual void sendDataToUav(cMessage* msg);

virtual void sendDataToCH(cMessage* msg);
```

3.7 Results:

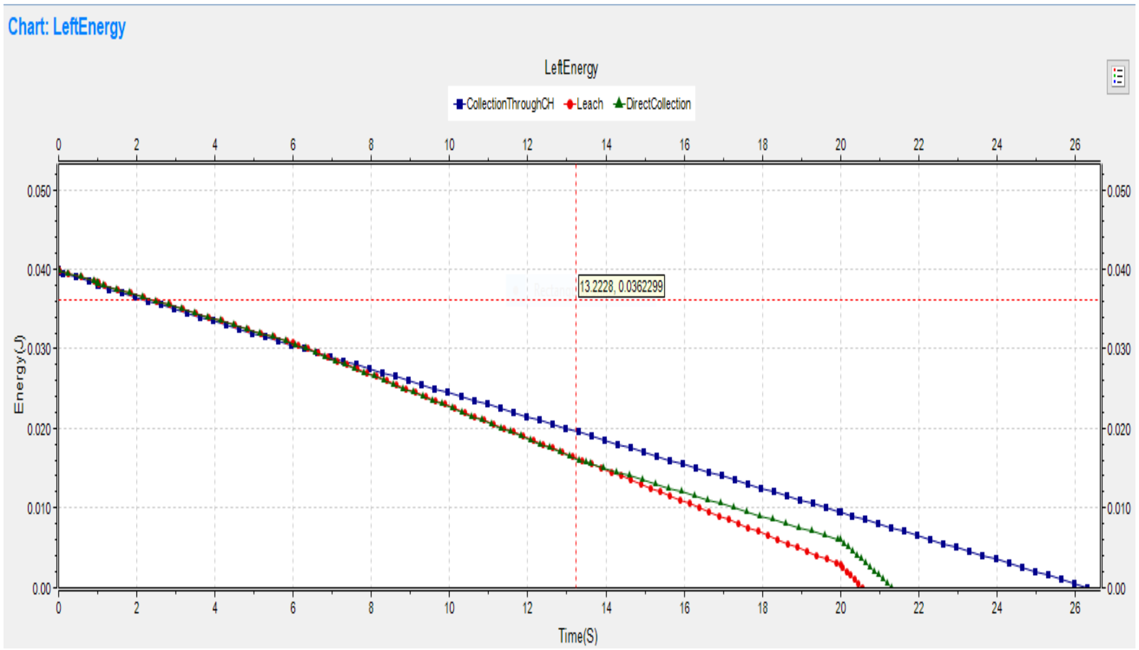

Fig3.5 : Simulution result comparision.

3.8 CONCLUSION:

in this chapter,we focused to compare residual energy level of our two proposed solution with standard leach protocol,And we in included simulation results in omnet++ simulator program.

General Conclusion :

The goal of the thesis was to reduce energy consumption of data collection in wireless sensor networks and that by implementing LEACH protocol on flying nodes and ground nodes.

The data that was exchanged in this experiment was videos this make the energy was consumed very quickly,thus we proposed two approaches.

First approach the communication was among the flying nodes and ground nodes directly in which each node sends the recorded video to its flying node that belongs to him.

Second appoach the communication was among the flying nodes and elected ground nodes ,in which the elected ground nodes gather data from his memebers than they aggregate the videos then send the data to flying nodes that belongs to him.

In the last of thesis we included the result of implementing our proposed approaches compared to standard solution ,And there was improving in energy consumption.

Bibliography:

[1] OMAR SAMI OUBBATI, ABDERRAHMANE LAKAS, FEN ZHOU, MESUT GÜNE¸S, MOHAMEDBACHIR YAGOUBI, A SURVEY ON POSITION-BASED ROUTING PROTOCOLS FOR FLYING AD HOC NETWORKS (FANETS).

[2] AMIRACHRIKI, HAIFATOUATI, HICHEMSNOUSSI, FAROUKKAMOUN, COMMUNICATION, MOBILITY MODELS AND SECURITY ISSUES.

[3] OZGUR KORAY SAHINGOZ, NETWORKING MODELS IN FLYING AD-HOC NETWORKS(FANETS): CONCEPTS AND CHALLENGES, J INTELL ROBOT SYST (2014) 74:513–527

[4]] P. P. KAMBLE. WIRELESS SENSOR NETWORKS FOR EARTHQUAKE DETECTION AND DAMAGE MITIGATION SYSTEM. ARTICLE,INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN COMPUTER AND COMMUNICATION ENGINEERING.

[5] H. KARL AND A. WILLIG. PROTOCOLS AND ARCHITECTURES FOR WIRELESS SENSOR NETWORKS, BOOK

[6] I. F. AKYILDIZ, W. SU, Y. SANKARASUBRAMANIAM, AND E. CAYIRCI. A SURVEY ON SENSOR NETWORKS.ARTICAL , IEEE COMMUNICATIONS MAGAZINE

[7]] RAKOCEVIC AND GORAN. OVERVIEW OF SENSORS FOR WIRELESS SENSOR NETWORKS. TRANSACTIONS ON INTERNET RESEARCH, 2009.

[8] J. POLASTRE, J. HILL, AND D. CULLER. VERSATILE LOW POWER MEDIA ACCESS FOR WIRELESS SENSOR NETWORKS. IN : PROC. 2ND INT'L CONF. EMBEDDED NETWORKED SENSOR SYSTEMS (SENSYS'04), PAGES 95–107, 2004.

[9] V. RAJENDRAN, J. J. GARCIA-LUNA-AVECES, AND K. OBRACZKA. ENERGY-EFFICIENT, APPLICATION-AWARE MEDIUM ACCESS FOR SENSOR NETWORKS. IN IEEE INTERNATIONAL CONFERENCE ON MOBILE ADHOC AND SENSOR SYSTEMS CONFERENCE (MASS'05), 2005.

[10] I. RHEE, A. WARRIER, M. AIA, AND J. MIN. Z-MAC : A HYBRID MAC FOR WIRELESS SENSOR NETWORKS. IN : PROC. 3RD INT'L CONF. EMBEDDED NETWORKED SENSOR SYSTEMS (SENSYS'05), PAGES 90–101, 2005.

[11] W. R. HEINZELMAN, J. KULIK, AND H. BALAKRISHNAN. ADAPTIVE PROTOCOLS FOR INFORMATION DISSEMINATION IN WIRELESS SENSOR NETWORKS. IN PROCEEDINGS OF THE 5TH ANNUAL ACM/IEEE INTERNATIONAL CONFERENCE ON MOBILE COMPUTING AND NETWORKING, MOBICOM '99, PAGES 174–185

[12]] C. INTANAGONWIWAT, R. GOVINDAN, AND D. ESTRIN. DIRECTED DIFFUSION : A SCALABLE AND ROBUST COMMUNICATION PARADIGM FOR SENSOR NETWORKS. IN PROCEEDINGS OF THE 6TH ANNUAL INTERNATIONAL CONFERENCE ON MOBILE COMPUTING AND NETWORKING, MOBICOM '00, PAGES 56–67

[13] C. SCHURGERS AND M. B. SRIVASTAVA. ENERGY EFFICIENT ROUTING IN WIRELESS SENSOR NETWORKS. IN 2001 MILCOM PROCEEDINGS COMMUNICATIONS FOR NETWORK-CENTRIC OPERATIONS : CREATING THE INFORMATION FORCE (CAT. NO.01CH37277), VOLUME 1, PAGES 357–361,

[14]] 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, WSNA '02, PAGES 22–31,

[15] R. C. SHAH AND J. M. RABAEY. ENERGY AWARE ROUTING FOR LOW ENERGY AD HOC SENSOR NETWORKS,ARTICLE.

[16] W. R. HEINZELMAN, A. CHANDRAKASAN, AND H. BALAKRISHNAN. ENERGY-EFFICIENT COMMUNICATION PROTOCOL FOR WIRELESS MICROSENSOR NETWORKS.BOOK.

[17]] N. BULUSU, J. HEIDEMANN, AND D. ESTRIN. GPS-LESS LOW-COST OUTDOOR LOCALIZATION FOR VERY SMALL DEVICES,ARTICLE.

[18] Y.-B. KO AND N. H. VAIDYA. LOCATION-AIDED ROUTING (LAR) IN MOBILE AD HOC NETWORKS,BOOK.

[19] S.-M. SENOUCI AND T. M. RASHEED. MODIFIED LOCATION-AIDED ROUTING PROTOCOLS FOR CONTROL OVERHEAD REDUCTION IN MOBILE AD HOC NETWORKS,,ARTICLE.

[20] T. HE, J. A. STANKOVIC, C. LU, AND T. ABDELZAHER. SPEED : A STATELESS PROTOCOL FOR REAL-TIME COMMUNICATION IN SENSOR NETWORKS. IN PROCEEDINGS OF THE 23RD INTERNATIONAL CONFERENCE ON DISTRIBUTED COMPUTING SYSTEMS.

[21] C. DIALLO, M. MAROT, AND M. BECKER. LINK QUALITY AND LOCAL LOAD BALANCING ROUTING MECHANISMS IN WIRELESS SENSOR NETWORKS. IN 2010 SIXTH ADVANCED INTERNATIONAL CONFERENCE ON TELECOMMUNICATIONS,

[22] C. DIALLO, M. MAROT, AND M. BECKER. LINK QUALITY AND LOCAL LOAD BALANCING ROUTING MECHANISMS IN WIRELESS SENSOR NETWORKS. IN 2010 SIXTH ADVANCED INTERNATIONAL CONFERENCE ON TELECOMMUNICATIONS.

[23] E. M. BELDING-ROYER. MULTI-LEVEL HIERARCHIES FOR SCALABLE AD HOC ROUTING. WIRELESS NETWORKS,ARTICLE.

[24] J. SUCEC AND I. MARSIC. HIERARCHICAL ROUTING OVERHEAD IN MOBILE AD HOC NETWORKS,ARTICLE.

[25] P. JACQUET, P. MUHLETHALER, T. CLAUSEN, A. LAOUITI, A. QAYYUM, AND L. VIENNOT. OPTIMIZED LINK STATE ROUTING PROTOCOL FOR AD HOC NETWORKS. IN PROCEEDINGS. IEEE INTERNATIONAL MULTI TOPIC CONFERENCE.

[26] C. E. PERKINS AND P. BHAGWAT. HIGHLY DYNAMIC DESTINATION-SEQUENCED DISTANCE-VECTOR ROUTING (DSDV) FOR MOBILE COMPUTERS.

[27] J. J. GARCIA-LUNA-ACEVES AND M. SPOHN. SOURCE-TREE ROUTING IN WIRELESS NETWORKS. IN PROCEEDINGS. SEVENTH INTERNATIONAL CONFERENCE ON NETWORK PROTOCOLS,BOOK.

[28] R. G. OGIER. EFFICIENT ROUTING PROTOCOLS FOR PACKET-RADIO NETWORKS BASED ON TREE SHARING. IN MOBILE MULTIMEDIA COMMUNICATIONS, 1999. (MOMUC '99) 1999 IEEE INTERNATIONAL WORKSHOP.

[29] C. E. PERKINS AND E. M. ROYER. AD-HOC ON-DEMAND DISTANCE VECTOR ROUTING. IN MOBILE COMPUTING SYSTEMS AND APPLICATIONS, 1999. PROCEEDINGS. WMCSA '99. SECOND IEEE WORKSHOP.

[30] S.-M. SENOUCI AND M. NAIMI. NEW ROUTING FOR BALANCED ENERGY CONSUMPTION IN MOBILE AD HOC NETWORKS. IN PROCEEDINGS OF THE 2ND ACM INTERNATIONAL WORKSHOP ON PERFORMANCE EVALUATION OF WIRELESS AD HOC.

[31] V. TOUBIANA, H. LABIOD, L. REYNAUD, AND Y. GOURHANT. PERFORMANCE COMPARISON OF MULTIPATH REACTIVE AD HOC ROUTING PROTOCOLS. IN 2008 IEEE 19TH INTERNATIONAL SYMPOSIUM ON PERSONAL.

[32] K. I. LAKHTARIA. ANALYZING ZONE ROUTING PROTOCOL IN MANET APPLYING AUTHENTIC PARAMETER,ARTICAL

[33] J. KLEINBERG. COMPUTING : THE WIRELESS EPIDEMIC. NATURE, 449 :287–288, SEP. 2007.

[34] C. WESTPHAL. OPPORTUNISTIC ROUTING IN DYNAMIC AD HOC NETWORKS : THE OPRAH PROTOCOL. IN 2006 IEEE INTERNATIONAL CONFERENCE ON MOBILE AD HOC AND SENSOR SYSTEMS, PAGES 570–573, OCT. 2006.

[35] D. S. J. D. COUTO, D. AGUAYO, J. BICKET, AND R. MORRIS. A HIGH-THROUGHPUT PATH METRIC FOR MULTI-HOP WIRELESS ROUTING. IN PROCEEDINGS OF THE 9TH ANNUAL INTERNATIONAL CONFERENCE ON MOBILE COMPUTING AND NETWORKING, MOBICOM '03, PAGES 134–146, NEW YORK, NY, USA, 2003. ACM

[36] Z. ZHONG, J. WANG, S. NELAKUDITI, AND G.-H. LU. ON SELECTION OF CANDIDATES FOR OPPORTUNISTIC ANYPATH FORWARDING. SIGMOBILE MOB. COMPUT. COMMUN. REV., 10(4) :1–2, OCT. 2006.

[37] M. LU AND J. WU. OPPORTUNISTIC ROUTING ALGEBRA AND ITS APPLICATIONS. IN IEEE INFOCOM 2009, PAGES 2374–2382, APR. 2009.

[38] https://www.lsec.icmc.usp.br/en/avens website, 24-09-2020

[39] https://inet.omnetpp.org/Introduction.html website, 24-09-2020

[40] https://omnetpp.org/intro/ website, 24-09-2020