IOT

KHALED ALNASER

POWERUNIT

* عندي 3 layers : IoT (low) و ال Fog وال Cloud .

* ال Middleware بتواصل بين ال IoT devices باخد من ال device الاول وهو بتواصل مع ال device التاني .

* هدول ال IoT Sensors ← low power و low cost (لهاي بيستقبل او يرسل رح يستهلك Energy).

* ال Fog layer ← high power و high cost و high Capabilities .

* بكون فيه اكتر من Fog و واحد منهم بيكون Fog leader . (كل Fog بمسك عدد من الـ IoT devices)

applications. Finally, few of the issues prevalent in the IoT field were explained and the future scope and applications were left out. A comprehensive focus on IoT forensic was presented by Yaqoob et al [37]. This work demonstrated novel factors that affect and enable forensics in the IoT domain. It further provided an investigation of several IoT forensics literatures and categorized them depending on sources of evidence, forensics phases, networks, enablers, forensics data processing, forensics tools, forensics layers, etc. to analyze their strengths and weaknesses. Several research challenges and issues were identified as future research directions. Sharma et al presented many definitions for IoT notions based on different researchers' perspectives [38]. The authors chronologically addressed the evolution of this technology. In the end, a slight discussion was provided to handle different IoT aspects such as its technology trends, communication standards, architecture and an overview of its future.

1.2 Findings and Impacts

There are many surveys that have been done to investigate different fields and issues of IoT domain till now. To the best of our knowledge, there are no prior surveys similar to ours. Interestingly, Table 1 displays how this work is distinctive from other highly cited papers mentioned in the previous section considering many perspectives out of which IoT paradigm, architectures, spreading spectrum techniques, layers protocols (original, recent, future enhancements), IoT middleware (recent challenges), IoT simulation tools, IoT applications, research security and challenges, and research history analysis and recommendations. In light of the aforementioned deficiencies of the related works, the major findings of this work can be summarized as follows:

1. Having higher value for researchers, as this survey is considered to be a starting point for their future researches because it gives the reader the opportunity to comprehend what has been done in IoT field, what still needs to be developed as well as what the risks and weakness factors are that need to be addressed. In addition, it exhibits the current trends in IoT research that are encouraged by the need for the convergence in multiple interdisciplinary technologies and IoT applications.
2. Highlighting diverse visions, definitions, and a thorough overview of IoT components and features for the reasoning of expediting a better comprehension of different IoT specifications by researchers and technicians.
3. Providing a detailed demonstration of different spreading spectrum techniques (i.e., Direct-Sequence Spread Spectrum (DSSS), Frequency-Hopping Spread Spectrum (FHSS), Chirp Spread Spectrum (CSS), Time-Hopping Spread Spectrum (THSS)). Based on such important information, the network designers can use the proper or suitable spreading spectrum techniques in their IoT communication systems, which will reduce crosstalk interference, obtain less static noise and data integrity, reduce signals susceptibility to multipath fading, avoid signals interference, and guarantee security implementation by making IoT data signals hard to detect, intercept or demodulate.
4. Providing insights and deep synopsis of the most recent standards and protocols, which are classified based on different IoT stack architecture (i.e., application, transport, network, and data link layers), thereby making sure that the reader will be aware of the full picture of the original, current, and future enhancements of each protocol. Matter of fact, conducting comparisons between all protocols in each layer from different perspectives will help the researchers and technicians in deciding which one suits them more quickly in professional and organized manners without digging through precise details provided in standard specifications, sources, and Request for Comments (RFC).
5. Presenting a comprehensive overview of the emerging challenges and issues in the IoT domain in order to be tackled through future researches. In fact, after studying numerous IoT research papers we have come to the conclusion that most of the challenges and security issues emerge from the remarkable increase in data traffic, the huge variety of traffic types, diversity of IoT devices, great variances in data forms, heterogeneous networks, etc. All of these concerns have a dramatic effect on the performance and QoS of the IoT systems.
6. Detailing the most and recent trends and specifications of IoT middleware aspects. In other words, we make sure that the readers get a full understanding of the recent challenges and issues that face the middleware field, the diverse classification of middleware architectures, and differences of emerging middleware platforms for each type of architecture.
7. Introducing a comprehensive overview of IoT simulators that are currently available through classifying them into categories according to their functions and then conducting comparisons while considering prevailing needs and aspects. Besides, the major challenges raised through moving from simulating the Wireless Sensor Network (WSN) environment into IoT are highlighted, thereby allowing the developers to upgrade the current versions of these simulators to suit IoT environment's requirements.
8. Analyzing the IoT research history, utilizing Scopus database through 2011 to 2020, in a very professional manner which primarily includes both IoT stack and middleware architectures. In particular, as far as the former is concerned, we analyze the growth and diminishment of publications in the whole IoT stack which includes data link and communication protocols, network, transport, as well as application layers. In regards to the latter one, we analyze its publications' growth and diminishment considering actor-based, event-based, cloud-based, and service-based architectures bearing in mind that addressing the challenges and limitations of middleware architectures has to take the functional components as service composition, registration, and discovery and non-functional needs, such as ease of deployment, privacy, security, availability, reliability, timeliness, and scalability all into consideration. As a result, we provide recommendations that will certainly attract most IoT researchers.

1.3 Paper Outline

The remaining of this paper is organized as follows:

- Overview of the IoT paradigm (definition, functional blocks, basic components of smart devices) (section 2).

what we [illegible] amount of technologies [illegible] for IoT term, the subsequent definitions have [illegible]

- Definition 1: Objects have virtual personalities [illegible] them to communicate and connect with user contexts and social [illegible]
- Definition 2: Interconnected things that have active roles in what could be called [illegible]
- Definition 3: This expression consists of two words: Internet which is defined as the worldwide network of enormous number of networks depending on communication protocols standards, whereas the word Things refers to all objects that are connected to that network based on the same standards [41].
- Definition 4: The environment of IoT network composes of physical and virtual entities, where these entities turn into virtual things inside a cyber-world. These things are embedded with different abilities as sensing, analyzing and processing and self-management based on interoperable communication protocols and specific criteria, these smart things should have unique identities and virtual personalities [6].
- Definition 5: IoT notion is anything that can be accessed from anywhere at any time by anybody for any service through any network. Thus, IoT can be called as 6Anys [27].

2.2 IoT Functional Blocks

An IoT paradigm is composed of a number of functional blocks, which ease different functionalities of smart objects like sensing, actuation, identification, management, and communication. Figure 1 shows these blocks with brief explanations in the following bullet points [27]:

- **Device:** Smart devices are the main units of the IoT system, where they are able to perform many operations such as sensing, monitoring, control, and actuation activities. They are also capable of exchanging data with applications and other smart servers. Each IoT device must be prepared with many interfaces to enable it to connect with other smart devices, where it consists of interfaces for Internet connectivity, I/O for sensors, audio/video, memory, and storage. IoT devices are varied according to the application they are utilized for. These applications could be smartwatches, wearable sensors, automobiles، industrial machines, LED lights, etc. [27].
- **Services:** There is an enormous number of applications that utilize IoT techniques from office automation and home appliances to manufacturing lines and goods tracking etc. Thus, it is required to apply specific IoT services in order to enhance IoT application development and to speed up its implementation. These services can be classified into identity-related services, services for device modeling, information aggregation services, devices discovery, devices control, collaborative aware services, ubiquitous services, data analytics and data publishing [27] [42].
- **Management:** The main feature of the IoT device, which distinguishes it from traditional devices that can be managed and controlled using mechanical buttons or switches, is remote management with or without human intervention. Furthermore, these devices can exchange data between each other to take a suitable decision later on [27] [40].
- **Security:** The data of networks, specifically data of wireless networks, is vulnerable to a massive number of attacks as a denial of service, spoofing, and eavesdropping, etc. Thus, IoT system tries to mitigate these attacks via the implementation of many security functions like privacy, authorization, authentication, data security, content integrity and message integrity [27] [39].
- **Application:** The application layer provides IoT users with interfaces that enable them to monitor and control diverse aspects of IoT applications. Furthermore, they permit users to analyze and visualize the status of IoT system at any time and from anywhere to take suitable action [27].

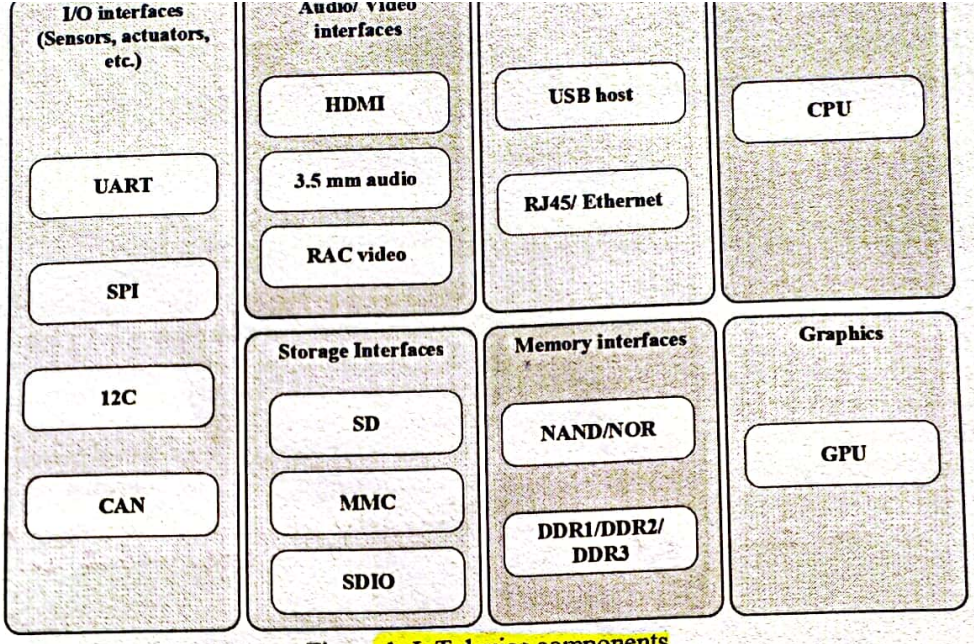


Figure 1: IoT device components

2.3 Basic Components of IoT Devices

IoT systems as mentioned before consisting of devices and applications, in order to allow them to communicate with each other they must have basic components, as will be illustrated below:

1. **Identification (ID):** Each object in IoT system must have a unique identification; an ID is assigned to an entity based on conventional parameters like universal product code, Media Access Control (MAC) ID, IPv6 ID or another custom method [27].
2. **Meta information:** Metadata consists of information about each device in IoT system such as device model, ID, revision, hardware, serial number and manufactured date.
3. **Security controls:** It resembles the "friend list" of Facebook, as the device owner can place restrictions on the devices types that can connect to his device [28].
4. **Service discovery:** This feature enables each IoT device to store details of all other smart devices that belong to the network in a specific directory. It is very important to keep these directories updated in order to get information about new devices that recently have joined the IoT network [28].
5. **Relationship management:** It allows each IoT device to start, update and terminate the relation between itself and other devices. Furthermore, it stores a list of the devices types that it should be connected with, according to the service type they are provided and based on human settings [27] [43]. For instance, a light sensor can create a relationship with a light controller device.
6. **Service composition:** This component enables interaction between smart objects and aims to provide users with the best-integrated service. To achieve such goals, the discovery service tries to find the required service that is provided by the smart object, to get benefit from it later on. It is also in charge of processing the data obtained from different objects to provide the user with the best solution [43].

3. Architecture of IoT

IoT connects millions of smart objects, which leads to more data traffic and the need for large data processors and storages [19]. Based on the above, IoT will face challenges regarding QoS, privacy, and security [44]. Thus, IoT architecture must take into consideration many issues such as interoperability, scalability, QoS, reliability, etc. [45]. In the literature, various IoT architectures have been suggested [46] [47] [48]. Nevertheless, each proposed architecture brings many shared drawbacks and fails to cover all of the IoT characteristics, which are summarized as follows [49]:

a. **Distributive:** IoT model is probably developed in an enormously distributed environment, where data can be collected from various sources and consequently can be processed via distinctive smart entities in a distributed procedure.
b. **Interoperability:** IoT devices that belong to distinct vendors have to communicate with each other to obtain mutual goals. Protocols and systems must be also designed in a manner that permits smart devices from numerous manufacturers to exchange their sensed data in an interoperable manner.
c. **Scalability**: Billions of objects are expected to join the network of any IoT environment. Thus, applications and systems that run on these environments must be able to manage and process a tremendous amount of data.
d. **Resources scarcity**: Both of computation units and energy are considered to be highly scarce resources.
e. **Security**: Users' feelings of being helpless and exposed under the control and dominant of an unknown external device could sorely handicap IoT deployment.

To overcome these issues, many researchers follow a specific-layered architecture for IoT infrastructure. In every proposed IoT architecture, similar techniques, functionalities, and services will be grouped into the same layer, which will facilitate the

development and enhancement of the architecture of each layer in the future [50]. There is no global consensus on the architecture of IoT, so different IoT architectures have been suggested by many researchers [49]. To the best of our knowledge and after an extensive search on IoT architecture models, we found that the superior model with respect to the elements that compose this environment is the "Three Based Architecture" model that is described in [51]. This architecture composes of the following three layers:

a. **IoT layer:** This layer contains all smart devices, entities, and end-users that are located in the IoT system.
b. **Fog layer:** All fog nodes are placed in this layer.
c. **Cloud layer:** All distributed cloud servers exist in this layer, where these servers consist of multiple processing units like a rack of high capabilities servers or it could be a huge server with multiple processing cores.

In every layer a set of nodes are grouped into domains, wherein a single IoT domain, that is composed of Nodes-Fog-Cloud agents, an application can be performed as depicted in Figure 2. The basic method that permits any IoT node, fog computing node and cloud server of communicating and interacting with each other is demonstrated as follows; firstly, an IoT node transmits its sensed data directly to a fog node that belongs to its domain application. As a result, the fog node processes the received data directly or sends it to another fog node or cloud server belongs to the same domain in order to send the reply back to the related IoT node. This step will reduce the service delay[5] of IoT node in receiving a response for any request, this comes from the location of the fog layer which allows its nodes to handle most of the requests come from the IoT layer [51]. The following sections demonstrate the architecture of each layer in the three-based architecture model.

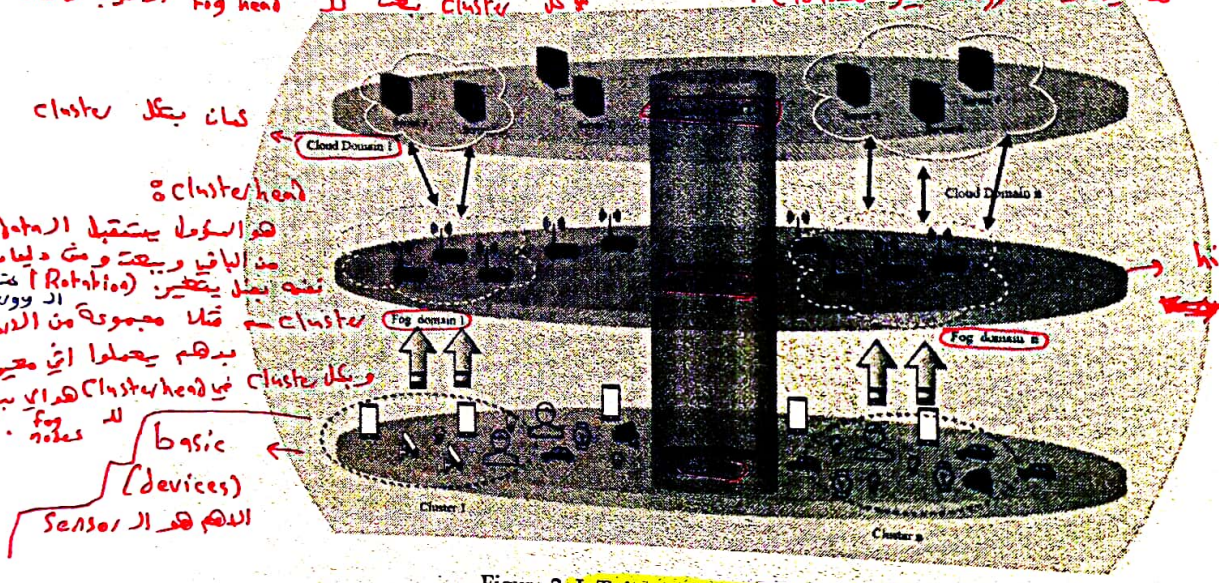


Figure 2: IoT three-based architecture layered

3.1 IoT Stack Architecture

Based on our thorough readings of a massive number of prior relevant surveys and books, we propose that the division of IoT stack consists of five layers which include perception, data link, network, transport, as well as application layers as shown in Figure 3, where all are discussed as follows:

i. **Perception Layer:** The primary mission of this layer is to sense the physical attributes of the entities that surround us and within the dominant of IoT network, where it depends on many sensing technologies such as RFID, WSN, Global Positioning System (GPS), etc. [49] [52]. Moreover, it is responsible for converting the sensed data to digital signals to be appropriate for network transmission. As a matter of fact, embedding intelligence and nanotechnology play an important role in this layer, as it enhances the processing capabilities of any object through inserting small chips (microcontroller) into smart devices that are used in everyday life [49].

ii. **Data Link Layer:** The IoT data link layer includes various communication protocols, which primarily provide services to the network layer. In fact, there are different standard technologies and protocols indicated by organizations for data link protocols out of which, Bluetooth, ZigBee, RFID, low power wide-area-networks, Z-wave, cellular [28].

iii. **Network Layer:** It is in charge of providing data with routing paths to be transmitted in packets form over the network area. The network layer establishes logical connections, delivers error reporting, manages and selects the routing path for data transmission. Moreover, this layer contains all network devices such as switches, firewalls, bridges, and routers, which are required to work along with suitable communication and routing protocols, such as 3G, 4G, 5G, Wi-Fi, infrared technology, ZigBee, fiber-to-the-x [49].

iv. **Transport Layer:** It works transitionally with the application layer to transmit and receive data without errors. The transmitting side of this layer is responsible for breaking messages that are received from the application layer into segments, and then send them to the network layer. In turn, the received segments will be reassembled into messages to be directly passed into the application layer by the receiving side. The transport layer provides features, such as packet delivery order, congestion avoidance, multiplexing, byte orientation, data integrity and reliability over the transmitted data.

[5] Service delay: Is the time period between the moment that IoT node transmits a service request and the time

Application Layer: This layer represents the front end of IoT architecture, where most of IoT potential will be exploited, because it provides IoT developers with interfaces, platforms, and tools that are required to implement IoT applications such as smart homes, intelligent transportation, smart health, and smart cities [49]. Moreover, it is responsible for receiving the processed data from the network layer.

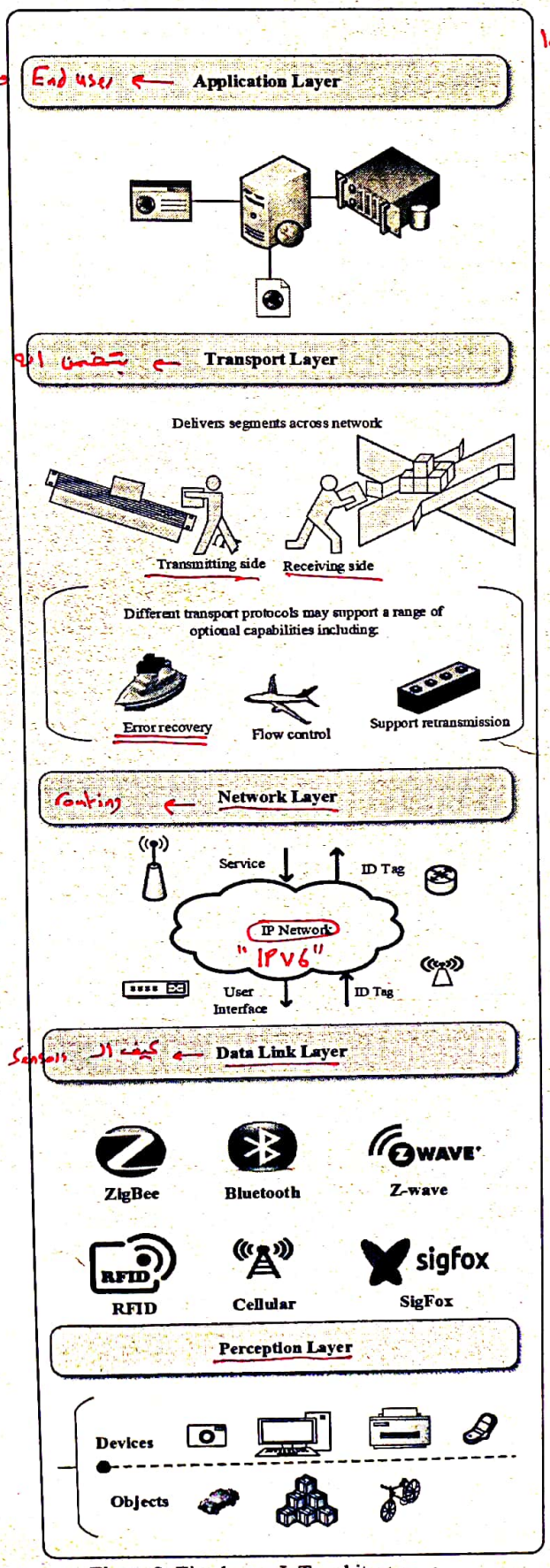


Figure 3: Five layers IoT architecture

3.2 Fog and Cloud Computing Layers

Big data that are generated by different IoT applications presents a new characteristic called Geo-distribution [53]. This new dimension requires that the sensed information has to be processed at the edge of the network area close to the smart devices instead of processing it by remote servers of cloud computing [54]. It is worth mentioning that it is indispensable to offer low latency response in order to allow smart objects to take the right action at the suitable time and to protect the integrity of sensitive infrastructure components. As a result, fog computing paradigm was suggested to extend cloud-computing services to the edge of IoT networks, to provide a highly virtualized platform that supplies many networking, storage and computational services between smart devices and cloud computing services [55]. Fog architecture comprises of four layers as depicted in Figure 4, which are monitoring, pre-processing, storage, and security layers [56].

3.2.1 Fog Layers Architecture

i. **Monitoring layer:** This layer is responsible for observing the activities of smart devices and networks. For example, it detects which sensor node performs some task, what task the node performs and at what time it is executed. Besides, this layer is in charge of monitoring the energy level of different network devices [28] [56].

ii. **Pre-Processing layer:** Performs data management, analyzing, filtering and trimming processes to generate useful and meaningful data.

iii. **Temporary storage layer:** After the pre-processing layer processes sensed data, it will be stored temporarily in the resources of this layer. The temporary storage layer offers many storage functionalities such as data storing, distribution, and replication [28].

iv. **Security layer:** It implements encryption and decryption techniques to protect the privacy and integrity of data.

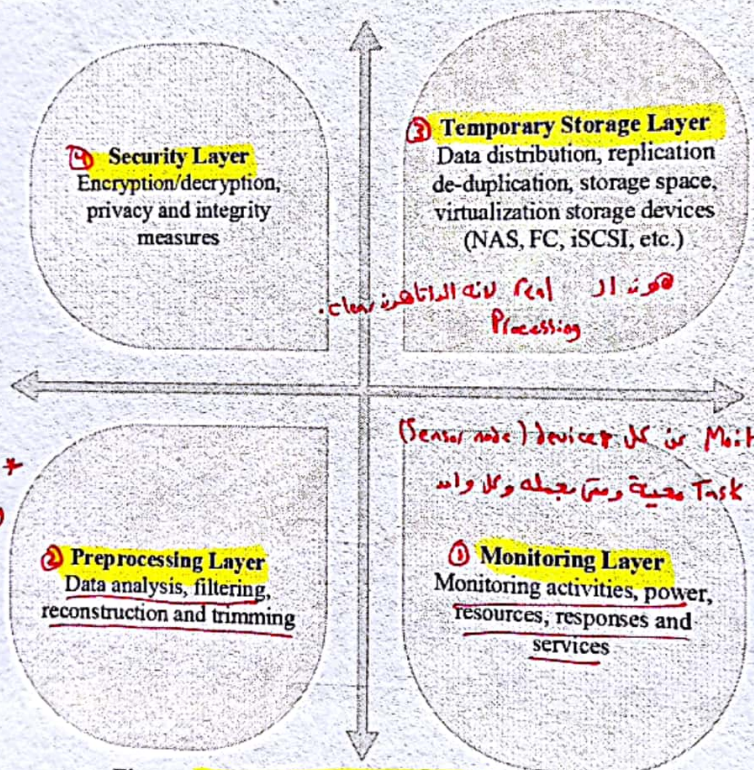


Figure 4: Layered architecture of fog computing

3.2.2 Why to Use Fog Computing Nodes

Fog computing nodes act as a bridge between smart objects, storage services, and large-scale cloud computing servers. This model extends network resources and services to the underlying network [56], so it has the capability of providing end-users with better delay performance services. Despite that, there is an important difference between the cloud and fog computing paradigms, where the cloud has enormous computational, communication and storage capabilities compared with fog computing [57], Figure 5 shows the roles of cloud computing and fog computing in the delivery of IoT services [58].

Connecting a massive number of smart objects to the internet such as smartphones, PCs, animals, and humans tracking, creates what is called the "Big Data" term that needs high capabilities to be stored, processed and analyzed. Fog computing nodes provide end-users with such abilities and are the best choice for many applications rather than farthest cloud computing for the following reasons:

1. Edge location, low latency, location awareness: According to that, fog computing provides its clients with rich applications and services with low latency requirements [57].
2. Geographical distribution: Applications and services that are hosted and processed by the fog nodes require widely distributed deployment of these nodes closer to the end-user. Fog, for instance, plays an essential role in delivering quality streaming to vehicles via access points and proxies that are positioned along tracks and highways.

3. Mobility supporting: It is common that fog applications communicate directly with mobile smart entities. Thus, fog computing is able to support mobility standards such as locator identifier separation protocol [28] [59] [60].
4. Real-time interactions: It has the ability to implement real-time interaction services since it can give an instantaneous response.
5. Dominance of wireless access.
6. Supporting online analytic and interaction with the cloud, as it plays a significant role in the ingestion and processing of a massive amount of data that are received from close smart devices.
7. Scalability: Fog permits IoT environments to grow, so as the number of smart devices increased, as a result, the number of fog nodes will be increased too to handle the new load. Such resource expansion cannot be achieved from the cloud side since the deployment of new servers is highly cost.
8. On the fly analysis: Fog resources aggregate data to transmit it partially processed to the cloud servers for additional processing.
9. Power constraints: Since most of the smart devices are battery-powered, long-distance communication toward the cloud will deplete their energy faster.

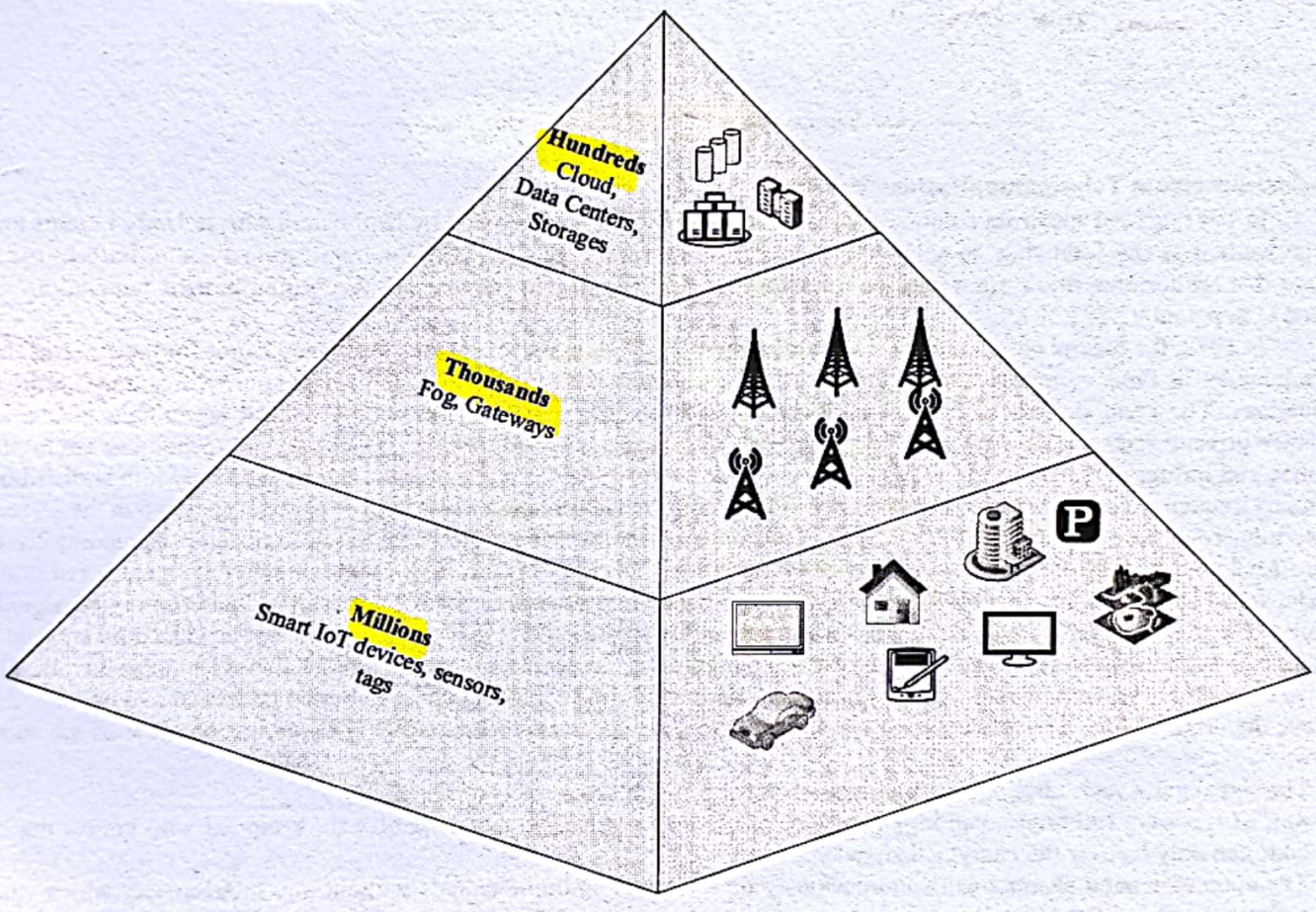


Figure 5: The role of the cloud and fog computing in the delivery of IoT services

3.2.3 Cloud Computing Architecture

In the IoT model, communication and information systems are embedded in a smart environment that surrounds us. This will lead to the generation of a massive amount of data that needs to be presented, processed and stored in an efficient, seamless and easy interpreting manner. According to [12], cloud-computing technology is the latest paradigm that proves its efficiency, scalability, autonomy, and reliability, as it provides high capabilities in dynamic resources discovery, ubiquitous access and composability[6], which are important for the prosperity of the future of IoT applications [49]. This platform plays several roles such as a data receiver from smart devices, a computer that analyzes and interprets distinct types of data, and as a supplier of web-based visualizations [61]. Many researchers try to construct a compatible architecture that can describe the function of the cloud computing paradigm as shown in Figure 6. This model consists of three layers, which are; the base layer that includes a database to keep details of all smart devices in the IoT network. The next layer is the component layer, which includes the codes that are required to interact with all IoT entities and employ a subset of these entities to execute a service or to query their status, where the last layer in this model is the application layer, which is in charge of providing users with the needed services [28].

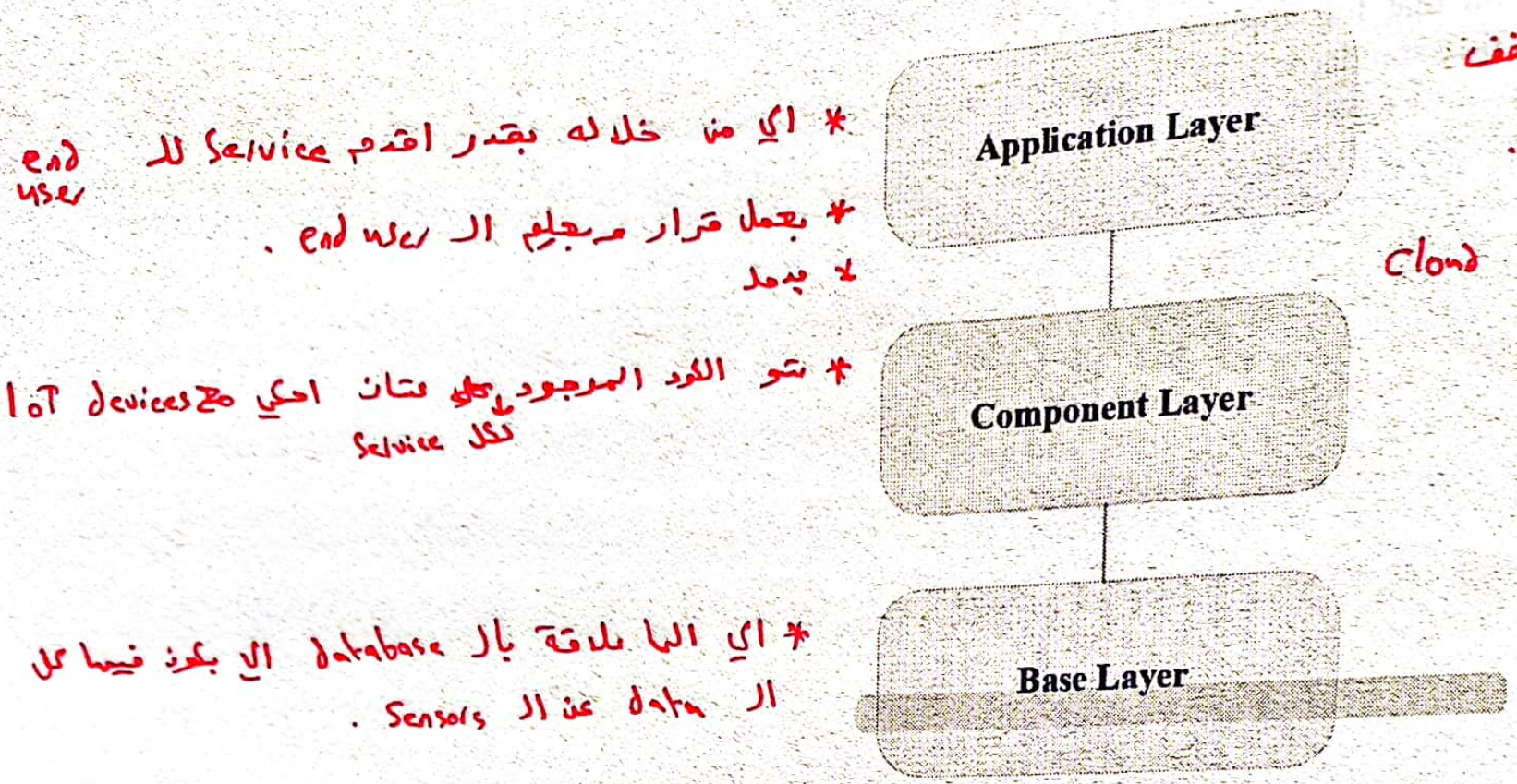


Figure 6: Cloud computing architecture

* لما اشتري الـ IoT device لازم اعرف نتو الـ Communication Protocol ولتو حستخدم الـ device لتعرف نحدد افضل Protocol.

→ Protocols to Communicate between devices.

Data Link Layer IoT Protocols / Communication protocols:

The following bullets summarized the most common IoT communication protocols:

(1) **Near Field Communication (NFC) protocol** [52] [53] [54]: is a very short range wireless communication protocol, through it mobile objects can communicate with each other over a few centimetres. All varieties of data can be transmitted between NFC devices, in seconds, through bringing them near to each other. This protocol is based on RFID, as it utilizes the alteration in magnetic field to allow devices communicate with each other. NFC devices can operate in two modes which are active and passive, in the active mode all communicating devices should generate magnetic fields, while in passive mode one of them generates magnetic field and the other utilize load modulation to transmit their data. The passive module is very useful in communication between power constrained devices in order to optimize energy. All of smartphones today are NFC enabled. The following are some parameters values of the NFC protocol.

هون ما في داعي يكون في Internet, بتواصلو الـ devices آما مح بدفش.

* المهم نعرف الخصائص لانواع الـ Communication Protocols

- *Frequency: 13.56MHz*
- *Range: 10cm* → For Communication
- *Data Rates: 100–420kbps*

(2) **Low-power wireless Personal Area Network (6LowPAN) protocol**: this protocol was developed to permit wireless smart things to connect through the internet via employing IPV6 functionality, by taking the nature of wireless IoT networks into consideration and constructing very compact message header format [55].6LoWPAN breaks down hindrances to utilize IPV6 addressing protocol in limited processing capabilities, low power [56]and low data rate IoT objects over limited bandwidth of wireless networks [11] [57] [58].

(IPV6) بتواصلوا الـ IoT devices مع الانترنت مثل هيك هوائيا

- *Frequency: 2.4GHz*
- *Range: -* 10 - 100 meters → For communication
- *Data Rates: 20,40,250 kbps*

(3) **Bluetooth Low Energy (BLE) protocol:** is an emerging wireless technology that was developed via the Bluetooth Special Interest Group (SIG) [59] for short range communication. It has been developed as a low-power solution for monitoring and controlling applications. This protocol supports quick transmission of data packets with data rate up to 2Mbps in Industrial Scientific Medical (ISM) band. There are two types of devices that implement BLE protocol: master and slave. Where master acts as a prime device that can connect to several slaves. Let us assume an IoT scenario in which a PC or a phone act as a master, where the other devices like fitness tracker, smart watch or thermostat act as slaves. In such scenario slaves have to be very power efficient, so they will be in sleep mode until they receive packets from the master device [11].

* لما بدي ابحث لمنطقة ازيد الـ Power بزيد الـ Transmission فبصير الـ Range اطول لكن بالاصل هو المقياس للـ Communication Range

- *Frequency: 2.402 – 2.481 GHz*
- *Range: up to 100 meters*
- *Data Rates: 125 Kbit/s, 1 Mbit/s ,2 Mbit/s*

(4) **ZigBee:** this protocol was developed to suit wireless communication of personal area networks (PANs), as it offers high security, low-power operation, scalability and robustness [60].This technology will be embedded in vast range of applications and products across commercial, consumer, government and industrial markets [61]

- *Frequency: 2.4GHz*

highly Recommended + بقدر استخدمه عالـ Internet لكن بستخدم اكتر بالـ Domain IoT لانه اللغة الـ 6LowPAN ابعد.

* الفرق بين الـ transmission range والـ Communication Range: قريبين من بعض بس المقياس الرئيسي لقدية استقبل data/packets هو الـ Comm Range, في حال بدي يكون الـ trans Range اكبر فبقوي الـ Power ليزيد.

* اي اني بقدر استخدمه عالانترنت.

- *Range: 10-100m*
- *Data Rates: 250kbps*

(5) Radio Frequency Identification (RFID) protocol: it is a low power, low cost wireless communication protocol, that consists of passive or battery assisted passive (BAP) small chips with antennas that are named tags [11]. These tags can send data only when they are powered through electromagnetic field generated by the reader. The life time of RFID tags can be measured in decades, as they do not depend on an internal source of energy to operate [62] [63],which makes this technology suitable in many applications. None the less, the primary hurdle of this technology that the RFID tags can operate only under the reader coverage domain which is not more 10 m in fully passive tags, while its range reaches 50m in BAP tags [64].

- *125 – 134 KHz – Low Frequency (LF) – An extremely long wavelength with usually a short-read range of about 1 – 10 centimetres.*
- *13.56 MHz – High Frequency (HF) with read range of about 1 cm - 1 m. require a long-read range.*
- *865 – 960 MHz – Ultra High Frequency (UHF) – A short, high-energy wavelength up to 30* [65].

(6) Low Power Wide-Area-Networks (LPWAN) protocols: are low power, low cost and low-bandwidth communication over long distances. Moreover, they transmit over sub-GHz radio frequencies from 433MHz to 868 MHz in Europe and up to 915 MHz in USA, with transmission ranges from 1 m up to 50Km [66].As many industrial, civil and other IoT applications need to operate over a wide area, then existing technologies which operate in the frequency bands of 2.4 or 5 GHz ISM (industrial, scientific and medical area) are required. To overcome this problem, a number of low power wide-domain networking protocols have arisen. The following are the characteristics of LPWAN protocols:

- The devices that implement these protocols, are very low power consumption.
- These protocols support the transmission of short packets only, commonly 100 bytes or less.
- The devices that implement LPWAN protocols consist of very low-cost unit, these devices usually cost less than few dollars.
- These devices are designed to have good coverage inside and outside their domains.

The sections below discuss the current cutting-edge approaches for LPWAN technologies.

6.1. LoRaWAN: is a physical layer communication protocol, with low power consumption and long battery life time that reaches up to 10 years. LoRaWAN targets wide-area network (WAN) services and applications (such as M2M, industrial applications and smart city) [66] that require long communication distances, which have ranges from (2-5) Km in urban territories and up to 15Km in suburban areas [67].It support large networks with billions of devices ,thus the data rate of this protocol ranges from 0.3 kbps to 50 kbps. The communication through LoRaWAN is bidirectional.

- *Frequency: Various (100Hz, 869 MHz (Europe) and 915 MHz (North America))* [67]
- *Range: 2-5km (urban environment), 15km (suburban environment)*
- *Data Rates: 250 bps–5.5 kbps/11 kbps/50 kbps.*

يستخدم لو بدهم Sensors يحكوا
مع بعض بـ Range عالي لغاية
عالي Data Rate و 1Km

6.2 Low Power WIFI (WIFI HaLow- IEEE 802.11ah) protocol: is a wireless communication MAC and PHY layer protocol, that was developed to communicate with wireless sensor networks applications, that requires long-range communication and less power consumption.

- *Frequency: 900 MHz*
- *Range: up to 1 km.*
- *Data Rates: up to 347 Mbit/s.*

6.3 IEEE 802.15.4g (marketed as WiSUN) protocol: this protocol operates in both sub-GHz bands and 2.4 GHz bands. Data rates varies from (40 -1000) kbps. Furthermore, it supports PHY packet sizes starting from 1500 bytes. Also, this protocol enables IP packets to deliver without fragmentation.

ميزته الفائدة

* ممكن نستخدمه للانترنت بناء على خصائصه (لانه مسافة طويلة rate عالي وما خبي Fragmentation).

- *Frequency: ISM <2.4GHz*
- *Range: up to 1000 meters.*
- *Data Rates: 50 kbps to 1Mbps.*

6.4. SigFox protocol: is a narrowband (or ultra-narrowband) technology, that is developed for connecting a massive number of power constrained devices. This protocol operates on 868-MHz frequency band, the spectrum is divided into 400 channels of 100Hz [68]. IoT devices can transmit up to140 packets each day with data rate up to 100 bps and with coverage range of (30–50) km in rural territories and (3–10) km in urban territories [69].

انعمل عتان الـ devices اي الها Power قليلة جدًا نشبكا عليه.

- *Frequency: 868-MHz.*
- *Range: 30-50km (rural environments), 3-10km (urban environments).* → Amazing Range
- *Data Rates: 10-100bps.*

اعلى واحد صار لحد الان.

لما كل ما كبر الـ Range بقل الـ DataRate عتان ما يستهلك Power عالي.

ممكن ما نشبكا عالانترنت

(7) **Neul:** this protocol operates in sub-1GHz band, it uses small chunks of the TV White Space spectrum, in order to transmit data with high coverage, high scalability, low-cost and low power in WSNs. It transmits over UHF band with a range of 300 MHz to 3 GHz, this communication technology is called Weightless[1] [11]. (For wireless sensor network environment)

ما بنشتبك عالنت عتان المسافة طويلة ضفي noise عالي.

- Frequency: 900MHz (ISM), 458MHz (UK), 470-790MHz (White Space)
- Range: 10km
- Data Rates: Few bps up to 100kbps

اكثر من هيك بزيطش (amazing Protocol)

(8) **Z-Wave:** it is a low power wireless communication technology, that is mostly designed for domestic automation products, such as smart light controller and other sensors inside home devices. This technology aims to provide reliable communication with low latency transmission of small data packets, and small data rates that can reach up to 200kbps [70]. Z-wave radio operates in 900 MHz ISM bands. Once it is deployed on a control device, then it can enable the controlling of up to 232 smart devices [71].

تستخدم بال Smart homes.

- *Frequency: 868 MHz in Europe and 908 MHz in the United States 900MHz (ISM)*
- *Range: 30m* → جوا البيت كافي
- *Data Rates: 9.6/40/200kbit/s*

[1] ***The Weightless*** protocol is a new standard for machine to machine(M2M) communications, that is based on white space wireless technology. This technology was developed to meet the requirements of numerous applications, with low cost, low power and good coverage range. Moreover, this protocol offers communications through low frequency spectrum and long rang, and data transmission over white space band and unlicensed spectrum. The name weightless reflects the light-weight nature of this protocol, since the transmitted data is diminished for devices communicated with a few bytes [174] [175].

مشكلته الـ Power لانه الـ Range عالي والـ Data Rate عالي.

(9) **Cellular:** any IoT service that demands to operate over long distances, can benefit from deploying GSM,3G and 4G cellular communication protocols. As their abilities of transmitting high quantities of data packets, particularly in 4G technology. Based on that, the communication through cellular protocols is so expensive, and extremely power consumption for many applications [72].

- *Frequencies: 900/1800/1900/2100MHz*
- *Range: 35km to 200km* "اعلى اشي وصلناه"
- *Data Rates (typical download): 35-170kps General Packet Radio Service (GPRS), 120-384kbps (EDGE), 384Kbps-2Mbps Universal Mobile Telecommunications System 3G(UMTS), 600kbps-10Mbps High Speed Packet Access (HSPA), 3-10Mbps Long-Term Evolution 4G (LTE)* ← ولسا 5G اكتر كمان

ممكن تستخدم بس مش المدن الاساسي ← لاستخدم بالانترنت

(10) **Telensa:** this communication protocol transmits over Ultra Narrow band technology, sub 1GHz unlicensed ISM bands. In addition, it provides fully bi-directional communication, so it is convenient for monitoring and control operations. A Telensa sink node could connect up to 5000 devices, and its communication range can reach up to 2km in urban territories and 4 km in rural domains. The life time of a Telensa node can be reach up to 20 years [73]. This protocol is applicable for many applications such as smart parking, smart lighting and other smart city applications [74] [75].

- *Frequency: 868 MHz*
- *Data Rates: downlink rate of 500bps and an uplink rate of 62.5 bps*
- *Range: 1km.*

← ضواحي

الـ node الرئيسي ممكن يشيل 5000 nodes

مدن

يستخدم في الـ Smart city اللي بدها تعيش لفترات طويلة

Transmission range	500m -1 km	1 km (urban) 10 km (rural)	(30–50) km (rural) (3–10) km (urban)	30 m	(2- 20) km 1G (35-200) km 2G Rural: 500 km/h *t, suburban: 120 km/h *t, 10 km/h *t (3G) 500 km/h *t (4G)	20km (rural) 3km (urban)
Power\ current consumption	2 µA- 8 mA	(3-50) µA	500 mW - 4W/ (19-49) mA	~5mW	1800mA (2G) 800mA (3G) (1,000–3,500) mW 4G	100µW
Number of nodes per network	5000	55000, 100 K devices per cell	*	232 nodes	4,000 devices /km^2 (4G)	5000 lights per base station
Applications	Smart meters, smart city, smart agriculture	Electric metering manufacturing automation, retail point of sale terminals, smart city	Smart farming, status monitoring, asset tracking, smart building, pallet tracking for logistics	Smart home	Voice Calls (1G) Voice calls, browsing and short messages (2G) Video conferencing, GPS and mobile TV (3G) Wearable devices, high-speed applications and mobile TV (4G)	Street lighting, smart city, air quality, traffic monitoring, smart waste bin management, and smart meter
Data rate	50 kbps- 1 Mbps	(30-60) kbps 200 kbps	(10-100) bps	(9.6, 40, 200) kbps	2.4 kbps (1G) 64 Kbps (2G) 144 kbps-2 Mbps (3G) 100 Mbps - 1 Gbps (4G)	500bps downlink 62.5 bps uplink
Spreading technique	DSSS	DSSS	FHSS	DSSS	FHSS, DSSS, CDMA[40]	*
Applicable routing protocols	RPL	*	*	AODV, DSR[41]	AODV, DSR, GPSR[42]	RPL
Mobility	Yes	Yes	Yes	Yes	Yes	Yes
Cryptography	AES, certificates, HMAC[43]	AES, LTE encryption	AES-128	AES-128	Voice scrambling (1G) Authentication and 128-bit key per subscriber (2G) SNOW3G cipher, Rijndael cipher, KASUMI cipher and AES-128 (3G) EPS integrity algorithm (4G)	City-data guardian
References	[236] [237]	[238] [239]	[238] [240]	[24] [241]	[242] [243] [244] [245]	[212] [246]

6. Middleware

It is anticipated that the number of IoT devices will reach around 50 billion in 2020 [247]. This massive number of smart things that are connected to the internet, represents the so-called IoTs, aims to make the surrounding environment more intelligent [248]. Based on the above, the amount of the collected data in the IoT environment will be immense and will create considerable defiance for both industries and researches domains. One of the major challenges that IoT paradigm confronts is machine-to-machine communication, where this challenge forms a big concern in IoT systems because of an abundant number of the existing smart devices that do not follow the same protocols, as most vendors do not care about the compatibility of their products with other competitors' brands. One of the proposed solutions to solve this issue is to enforce universal standards, which is very hard to be applied, while another proposed solution is to implement middleware software to facilitate the communication process among these devices. Middleware can be defined as a software that offers interoperability between incompatible applications and devices, also it hides all the details of smart objects [249] [250]. Hence, it acts as a software bridge between the applications and the things, as it enables IoT systems to work efficiently with each other [12] [20] [24] [251]. There are numerous middleware solutions, either a proprietary or an open-source provided through companies, where most of these solutions are similar to each other. However, there are no guidelines or performance metrics that enable us to compare these solutions to each other [249]. According to that, many challenges face IoTs middleware as described below [28]:

i. **Programming abstractions and interoperability**: To facilitate collaboration and data exchange among heterogeneous devices, IoT middleware aids to permit distinct sorts of smart devices to interact easily with each other.

ii. **Device management and discovery:** This property allows IoT devices to discover all other devices and services that are located in their network domain. The infrastructure of the IoT environment is mostly dynamic since all newly joined devices must announce their existence and the services they provide. Therefore, IoT middleware requires being scalable and provides APIs in order to list all IoT devices, their capabilities, and their services. In addition, APIs have to provide the users with abilities to categorize the devices based on their capabilities, manage devices depending on their remaining energy, report problems in IoT devices to the users and perform load-balancing procedures among them.

iii. **Big data and analytics**: IoT sensors collect an enormous amount of data that requires to be analyzed by specific algorithms based on a data type. Also, some of the sensed data may be incomplete because of the flimsy nature of wireless sensor networks. Thus, middleware should consider this issue and extrapolate incomplete data by using a suitable machine-learning algorithm.

[40] CDMA: Code Division Multiple Access
[41] DSR: Dynamic Source Routing
[42] GPSR: General Packet Radio Service
[43] HMAC: Hash based Message Authentication Code

iv. **Privacy:** Most data that comes from IoT applications and services are related to human personal life. Thus, security and privacy issues have to be considered when transferring and processing them, which is required to build mechanisms that address these issues by middleware.

v. **Cloud services:** Cloud computing part is the most important layer of any IoT system because all of the sensed data will be stored and analyzed in a centralized cloud. Therefore, IoT middleware should be run smoothly in distinctive types of clouds and enables IoT users to gain the most benefits from the data collected through smart sensors.

vi. **Context detection:** IoT applications are classified into two types, which are ambient data collection applications and real-time reactive applications. In the first type, sensors collect data that will be processed later on offline to get reasonable information that will be used for the same scenarios in the future, while in the second type systems should make a real-time decision based on the sensed data.

6.1 Architecture of IoT Middleware

The current architecture of IoT middleware is classified into three types based on the services they provide as follows [252]:

1. **Service-Oriented Architecture (SOA) or Service-Based Solution:** In SOA users and developers are allowed to employ or add different types of IoT devices to be utilized as services [30] [253]. Figure 12 represents the architecture of SOA middleware, which consists of three layers: The Physical layer that contains actuators and sensors, the Virtualized layer, which consists of cloud and infrastructure servers that are responsible for performing different computational operations, and the Application layer that composes of all services and utilities. SOA is deemed to be a heavyweight and a very high performing middleware, where it can be implemented on the nodes that communicate with the cloud servers or on a powerful gateway that is placed between the cloud layer and IoT devices layer. Based on that, this type of middleware is not suitable to be implemented on resource-constrained devices and it does not permit device-to-device communication.

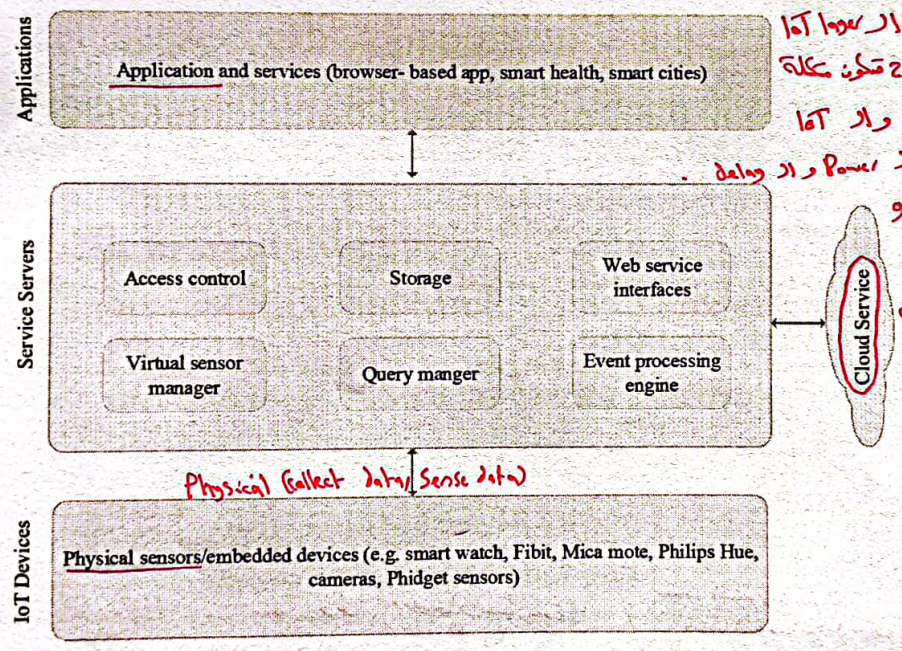


Figure 12: Service-based IoT Middleware

2. **Cloud-Based Solution:** In this type, users are constrained by the number and types of smart devices that can be connected to IoT applications. In addition, the sensed data can be easily collected and interpreted, because different used cases can be programmed and then determined in advance [30]. The resources of the cloud-computing environment restrict the operational components of this middleware. These functions such as storage system or computation engine are represented and managed by APIs, where IoT services are controlled and accessed by either cloud bolster RESTful APIs or by the applications provided by vendors as shown in Figure 13.
3. **Actor-Based Framework:** It is a lightweight middleware that can be implemented in Sensory, Gateway and Cloud Computing layers. The computational operations of this middleware are distributed in both sensory layer and mobile access layer as shown in Figure 14 [24].
4. **Event-Based Framework:** This type of middleware aims to improve the development of distributed systems by supporting the implementation of the publish\subscribe paradigm as shown in Figure 15. This paradigm is considered to be a communication infrastructure that aims to provides clients with general-purpose services, as it helps them to cope with the heterogeneity and complexity of large-scale and distributed environments. In event-based middleware, distributed application complexity is partially hidden from the programmer, which will, in turn, simplify the development and programming of many functionalities.

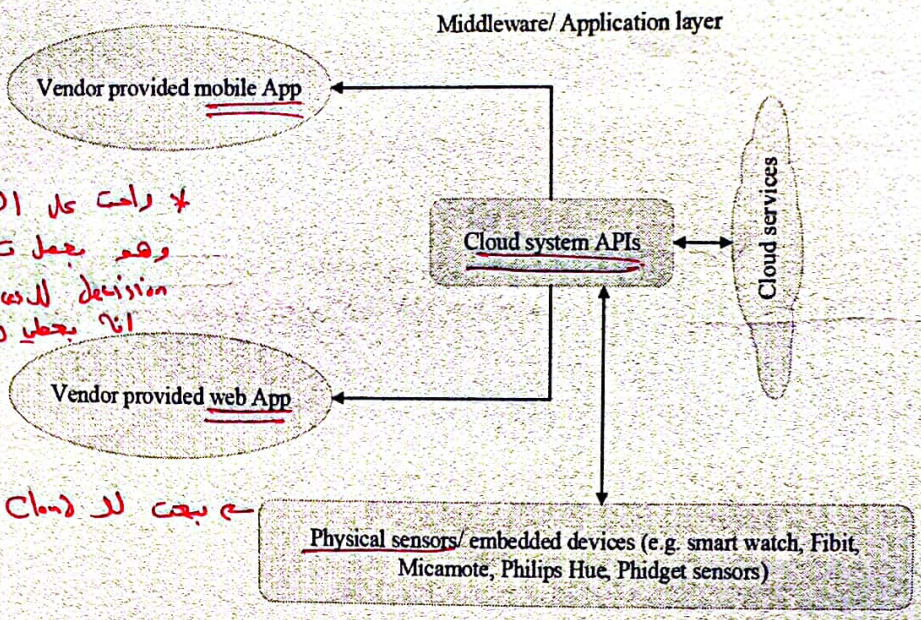


Figure 13: Cloud-Based IoT middleware

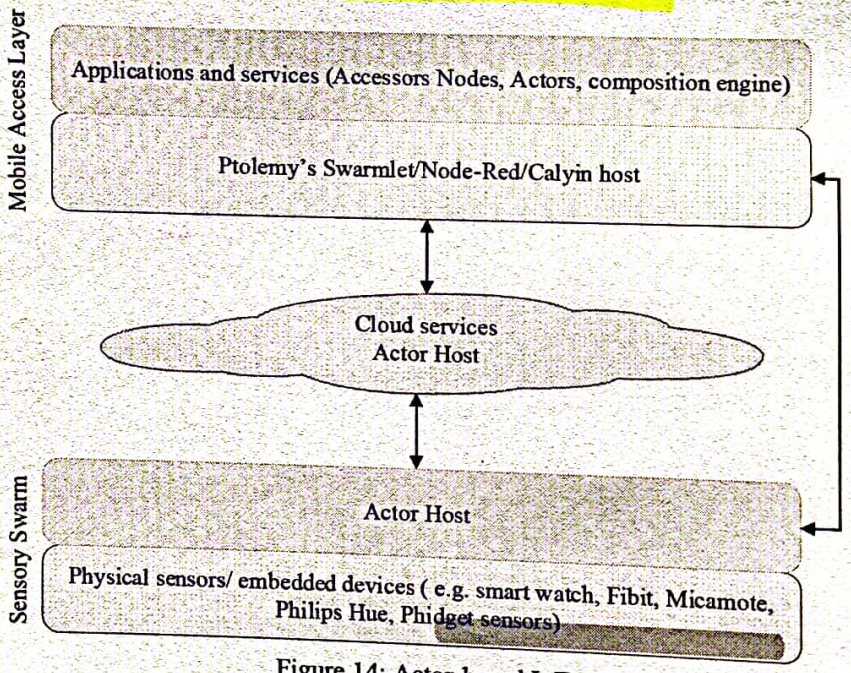


Figure 14: Actor-based IoT Middleware

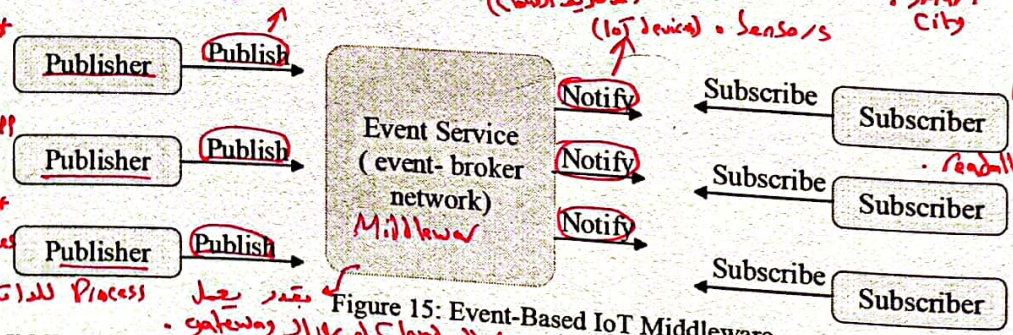


Figure 15: Event-Based IoT Middleware

6.2 Existing IoT Middleware Platforms

The following subsections summarize different solutions of IoT middleware based on its type, where Table 7 compare between IoT middleware platforms from different aspects.

6.2.1 Cloud-Based IoT Middleware

1. **AWS IoT:** This platform was developed by Amazon to manage cloud services, such as permitting millions of connected devices to interact securely and easily with other devices and cloud applications. AWS IoT allows customers to build their IoT applications in order to collect, process and analyze the sensed data to take a suitable decision without any need to manage

* مطلوب منا نعرف امثلة عن كل نوع . مفتقط اسماء

infrastructure by using AWS services like Amazon Kinesis and Amazon CloudWatch. Also, AWS IoT customers can keep track of all the devices that are communicating with their applications all the time [254].

2. **Azure IoT Hub:** It is a central platform that was released by Microsoft to manage bidirectional communication between IoT applications and their connected devices. Due to the high capabilities of Azure, it allows clients to construct full-featured, scalable IoT solutions with secure and reliable communications among the hosted cloud and a massive number of IoT devices. Azure IoT Hub supports various messaging patterns to control IoT connected devices, such as request-reply, file upload from devices and device-to-cloud telemetry [255].
3. **IBM Watson IoT:** This platform is built on the top of IBM Cloud to connect and control different IoT appliances, sensors industries, and home applications. IBM Watson provides its clients with an enormous set of adds-on, built-in tools, and Blockchain service that enable them to build their own IoT applications, manage their appliances, extract key performance indicators from their data, connect their tools and applications and process their collected data using historical and real-time analytics.
4. **Google Cloud IoT**: It is a fully managed service, which consists of a set of tools that provide a complete solution to securely and easily connect, process, manage, store, analyze, and visualize the generated data from dispersed devices, both in the cloud and at the edge of the network. Google Cloud IoT aims to have the ability to build models that can efficiently optimize and describe a client's business, anticipate problems, and improve operational efficiency [256].
5. **Xively:** It is a public cloud-based IoT middleware that provides a Platform as a Service (PaaS) [257]. This software aims to help companies and developers to connect, monitor and control distinctive types of IoT sensors [258]. Furthermore, it offers a web-based application that allows IoT devices to quickly connect and transmit data to its cloud servers. Also, it allows clients to retrieve their data from the cloud easily at any time and from anywhere, as it provides a time-series database that enables swift storage and retrieval of data [30].
6. **Oracle IoT:** It is a cloud-based service platform that enables users to build a real-time IoT solution, which can be integrated with enterprise applications, using robust security cloud capabilities, innovative and powerful edge analytics. Moreover, it processes the streaming of IoT data to merge insights into customer business easily and quickly. Oracle IoT permits clients to connect their devices to the cloud, which will help them in taking critical strategies and decisions [259].

6.2.2 Service-Based IoT Middleware

1. **LinkSmart (Hydra):** It is a web service platform that aims to eliminate the heterogeneity of distinctive devices and entities in the IoT environment [260] [261]. Furthermore, it enables controlling all types of smart devices regardless of their communication protocols, such as ZigBee, RF, RFID, Wi-Fi, Bluetooth, etc. LinkSmart distributes social trust computation and security units through middleware to make IoT devices and services more secure and trustworthy. A prime novelty of this middleware is supporting the utilization of IoT devices as services by embedding the required services in these devices. LinkSmart can also be used to manage specific IoT applications such as healthcare, agriculture and home automation. Also, it supports the self-configuration of devices and service discovery [262]. There are no local aggregation or processing units for the sensed data on IoT devices that implement LinkSmart, so it will be sent to the cloud to be processed and archived [263].
2. **Kaa:** It is an open-source platform that is managed by Cybervision Inc and KaaIoT technologies to enable building IoT solutions. Using web page Graphical User Interface (GUI), based on the Apache platform, enables the creation of data delivery schema, supporting multi-tenancy on servers and generation of endpoint Software Development Kit (SDK). Kaa enables interaction with endpoint devices directly or via gateways, while it secures their data by AES and RSA encryption methods.
3. **Global Sensor Networks (GSN)**: It aims to provide a uniform platform that supports adaptable deployment, sharing, and integration of heterogeneous IoT objects [20] [264]. This platform is built to meet the requirements of smart objects whether they are physical or virtual sensors or actuators. GSN is a Java platform that is deployed either on IoT cloud or servers, where a set of wrappers are permitted to feed the system with a collected live data, which will be processed later on based on XML specification files.
4. **ThingSpeak IoT:** It is an analytical open-source platform service that is developed by Matlab to enable communications between people and things. ThingSpeak provides users with tools that permit them to collect, visualize and analyze real-data streams in the cloud. Developers can easily store and retrieve data from devices and sensors by utilizing HTTP protocol over the internet [265].
5. **Aura:** This middleware is designed to ease the development of pervasive mobile IoT applications, by abstracting the differences among heterogeneous devices and permitting them to communicate with each other without any hindrances. Aura tries to optimize screen backlight and CPU to improve the performance level and reduce power consumption. Aura applies two concepts in interacting with events, where system layers reply directly to the upper layer in a proactive concept, while in a reactive concept all layers adjust their resources and performance based on demand [266].

6.2.3 Actor-Based IoT Middleware

1. **Calvin:** It is an open-source IoT platform that was developed by Ericsson to be implemented on the energy-constrained smart devices since it provides a portable and light-weight unified programming model, where its interfaces are defined via its input and output ports [267]. In Calvin, all low-level communication protocols of IoT devices are hidden as the communication between devices is performed through smart things' ports [30]. Moreover, Calvin can be implemented at the edge of IoT environments to reduce long-distance communications, which will minimize the latency and power consumption of IoT devices. The major merit of this middleware is its ability to migrate from one environment to another.

2. **Node-RED:** It is an open-source IoT platform that was developed by IBM and i be implemented at the edge of IoT network, because of its light footprint, wher based on an event-driven module and a non-blocking I/O is implemented. The n IoT applications easily through dragging and dropping the connected blocks that r of this platform are that it does not support service discovery and enables security
3. **Ptolemy Accessor Host:** This open-source platform was developed by Professor model embedded and real-time devices [269]. The main concept of this platform software components that interact and communicate with each other via messages
4. **Akka:** It is a set of open-source libraries and free actor-based platform that was applications using Java or Scala language. It permits users to meet business requ low-level codes, which will provide them with high performance, fault tolerance multi-threading behavior, abstracts the communication among applications and the clustered architecture [270] [271].

6.2.4 Event-based IoT Middleware:

1. **Hermes:** It is an event-based and scalable middleware that aims to ease the applications. Hermes creates self-managed event brokers based on P2P routing environments. It introduces a resilient solution against failures via automatic ada overlay broker network. Hermes middleware released two versions that sha implementation in distributed and large-scale applications and communications, brokers [272].
2. **Gryphon:** It is a patronizable publish\subscribe and highly scalable middleware th time data over the network. Gryphon is developed by Java interface to support web a publish\subscribe, and content-based multi-broker. This middleware contains algorithms, and an effective event matching engine. Also, it is based on an informati to specify the communication between the publisher and the subscriber.
3. **Rebeca:** This middleware is based on publish\subscribe technology to imple emphasizing on the design of efficient routing algorithms and employing profess Rebeca aims to prevent and reduce flooding the network by events by utilizing interoperability and subscription merging features with its services to support locatio Event scope function hides the details of service implementation, as transmission po interfaces among external and internal, and notification representation [272].
4. **FiWare:** It enables efficient, flexible, secure and scalable communications among was designed to support the control and monitoring of many IoT applications such [28]. This platform consists of many components such as APIs, reusable modules a build his IoT application. A set of sensed data collected by IoT sensors (context) is c on to a specific server called the broker. FiWare has developed API to query and stor is registered as a context consumer can retrieve the required data from the broker. Th called an adapter, where it is responsible for transmitting a particular type of contex

6.3 Open Research challenges of IoT Middleware:

Even though the IoT middleware field has handled many requirements and ... there are still some open challenges that require to be covered and solved. Th

- **Privacy and security**: Most of the middleware solutions restrict the app... and authorization, this is due to the resource-constrained devices in the IoT environment. Thus, privacy and security issue... need to be end-to-end and lightweight to suit the communication between cloud, gateway, and sensors.

7. Simulation tools of IoT Networks

Simulations are utilized to model system behavior at a certain time, where the simulation environment mimics and evaluate... a realistic scenario before building or implementing it in a real-life environment. Simulations are commonly used to estimate easi... the performance and cost effects on complicated systems. Using simulation tools to emulate IoT context is indispensable as ... supports assessing efficiently the performance of any application, because of the accuracy and the reliability of the results that a... provided. Diverse simulators have been built and proposed to mimic the behavior of mobile and distributed applications with sever... approaches, by making them compatible with many operating systems as Linux and Windows. However, every simulator has i... particular configuration requirements, which permits distinctive application aspects to be simulated. In general, any IoTs simulato... should offer high reliability when it simulates the scenarios that include heterogeneous sensors, provides computation or energ... efficiency estimation, supports scalability, and be able to support new requirements such as any new protocol [286]. Specifying ... suitable tool to simulate the IoTs environment is a challenging task since there are only a few simulation tools that have bee... designed for IoT applications. IoT simulators are classified according to the level of architectural layer and to the scope, they cove... into three categories [287]:

1. **Full Stack Simulators:** These simulators have been developed as a consequence of IoT revolution to provide users with th... ability to simulate IoT elements and devices. The main simulators in this category are Devices Profile for Web Servic... Simulator (DPWSim) and iFogSim [288] [289].

   - **DPWSim:** It is a cross-platform simulator that enables the development and the simulation of different IoT application... where the essential role of this platform is to create virtual IoT devices that can be discovered on IoT networks and ca... also communicate with each other through DPWS protocols [288]. Besides that, this simulator has a management tool tha... allows users to create, load, store, and manage their applications with high flexibility. The graphical user interface ... DPWSim is designed by Java language, which permits IoT users of interacting with their virtual environments smoothly... Finally, this toolkit helps in developing, prototyping and testing the DPWSim functionalities, but the main drawback ... this simulator that it has no support for new technologies and protocols [286].
   - **IFogSim:** This platform was emerged through upgrading and extending the capabilities of the CloudSim simulator [290... It allows the simulation of different IoT applications and the management of diverse resources that are distributed acros... the cloud and the edge of the network under various conditions and scenarios [289]. IFogSim permits users to evaluat... different resources management that is applicable in Fog environments according to their influence on energy consumption... latency, operational cost, and network congestion. Furthermore, it supports the simulation process of different types ... actuators and sensors by enabling the developer to build realistic network topologies.

2. **Big Data Processing Simulators:** These simulators concentrate on processing big data and evaluating the performance ... cloud resources, where the main simulators in this category are CloudSim [291], SimIoT [292], and IoTSim [293].

   Cloudsim: It is a toolbox utilized for modeling, experimenting, and simulating a cloud-computing environment... Developers and researchers can design a particular cloud system via this toolkit without any concern about low-lev... details of the cloud environments and the services they provide [291]. The library functions of the cloudsim is writte... using Java programming language and it consists of the main classes that are needed to mimic virtual machines, server... and clients to perform computational assets and to build applications. Furthermore, in order to set up a cloud environment... designers must utilize many simulation components such as virtual machines, data-centers, cloudlets, cloud coordinator... and data center brokers [294].

   - **SimIoT:** It is derived from SimIC simulator and has been developed to mimic large-scale resources management [29... [295]. SimIoT is used to estimate the time needed for processing data that is submitted either by IoT users or sensors to ... particular cloud, which is done by using numerous methods to simulate the communication between the cloud and Io... sensors [296].
   - **IoTSim:** This simulator was developed by [297] to simulate the behavior of IoT applications that are responsible fo... processing big data that is produced from various devices using the MapReduce framework. The vital contributions ... this simulator lie in allowing simulation and modeling of a network using virtual machines, permitting the processing ... IoT data through using big data framework (MapReduce), and supporting the IoT applications model.

3. **Network Simulators:** The growing of interest toward the field of WSNs has led to the booming of current simulators [298... The election process of a suitable simulator is a critical and time-consuming mission, particularly in the WSNs domain, sinc... there are many complicated scenarios and numerous protocols utilized in this domain that need specific features to exist in ... network simulator. Particular requirements of WSNs and the availability of a vast number of simulators make it difficult ... select a suitable simulator. Numerous WSNs simulators have been adapted to suit the simulation process of IoT environment... such as Cooja [299], QualNet [300], CupCarbon [301], OMNeT++ [302], and NS-3 [303].
   - **Cooja:** It is a discrete event and a flexible simulator, since several parts of Cooja functions can be extended or replace... by new functionalities such as OS, sensor node platforms, radio transmission models, and radio transceivers [298] [29... Cooja is developed and written in java language and runs over the Contiki operating system. However, this simulator ... not very efficient for many reasons as it requires a lot of calculations to deal with cross-level simulations, there is no G... interface, and the simulation process supports up to 10000 nodes only.

- **QualNet:** It is a tool that allows network designers to create a virtual scenario of all forms of video, data and voice networks. Any network scenario consists of nodes that represent WSNs elements and endpoints (switches, routers, ground stations, access points, mobile phones, satellites, firewalls, radios, servers, sensors, and other security equipment) and links that connect these nodes (Wi-Fi signals, internet circuits, LAN segments, LTE connections radio transmissions, etc.) [300]. The graphical user interface permits network designers to build their projects in 2D and 3D environments. Also, it allows the analysis of statistical data and packet tracing for debugging purposes [298].
- **CupCarbon:** It is an IoTs WSN and smart city simulator that aims to visualize, design, compile and validate the algorithms that are required for monitoring and collecting environmental data [304]. Furthermore, this simulator helps the researchers to test their wireless models and protocols. CupCarbon provides two simulation environments; the first one permits the generation of natural events like fires and it also supports the simulation of mobile entities such as flying objects and vehicles. On the other hand, the second simulation environment allows designers to represent discrete event scenarios of WSNs. Also, it grants WSNs designers the ability to simulate scenarios and algorithms in many steps as the following; a step for specifying designated nodes, another step to determine the communication types between these nodes, and finally determining routing to the base station. This simulator supports many IoTs communication protocols such as Lora, ZigBee, and WiFi.
- **OMNeT++:** It is a discrete event network simulator that is developed using C++ language by OpenSim company [302]. This simulator consists of GUI libraries for tracing, debugging and animating any network scenario. It also has graphical tools that enable building simulations and performing results computations. OMNet++ permits the hierarchical organization of any simulation scenario, because the number of layers is not restricted. The processes inside the virtual network such as drawing data flow charts, illustrating network graphics and displaying variables or objects during simulation are visualized through a graphical user interface [305]. The structure of the scenario is defined by using network description files (NED) that can be modified by the user via a graphical interface or a text file, where NED files are separated from the simulator to efficiently support the simulation of large topologies. Further, OMNeT++ is distinguished from other simulators in its ability to modify topologies in run time.
- **NS-3:** It is a discrete event simulator that is developed by C++ and Python language [286]. NS-3 permits researchers to analyze large-scale systems and different internet protocols in a controlled environment. This simulator has been improved to provide an open-source and an enormous network simulation platform, for the sake of supporting the education and the research in wireless networks. Concisely, NS-3 provides users with a simulation engine to conduct their simulation experiments and provide them with models that show how data packets perform and work. Furthermore, this simulator supports having multiple radio interfaces and channels for the same node [306]. Many wireless communication protocols can be implemented via NS-3 such as 802.15.4 and 6LoWPAN, but it does not support the protocols of the application layer [287].

To the best of our knowledge, there is no simulator that can be used to build a fully detailed representation of any IoT project until now. Consequently, to simulate a complete IoT project, multiple simulators should be used together such as data generation, big data processing, and packet tracing simulators. Table 8 shows a comparison between different IoT simulators based on popular IoT criteria and features, where the justification for each selected criterion is explained as follows:

- **Scope:** This criterion specifies the level of coverage for different architectural layers of IoT, where (IoT) means that the simulator has full coverage.
- **Last update:** It represents the time of the last maintenance or upgrading that is performed on the simulator.
- **Language:** It refers to the programming language of the simulator and reflects the portability degree of the simulated primitives to be used in subsequent hardware models.
- **Type:** It illustrates basic assumptions regarding the simulated entities and the relationships among them.
- **Layer of IoT architecture:** Represents the architectural layer(s) components, standards, and parameters that are supported by a specific simulator.
- **Evaluated scale:** The maximum network scale that can be simulated and provided through performing simulator evaluations.
- **Mobility:** Determines whether the simulator supports objects mobility or not.
- **Built-in IoT standards:** Specifies different protocols that are supported by a simulator.
- **Overall practicality:** It is a specific measure to indicate the utility behind simulating all components and services in the IoT environment.
- **Target domain:** Indicates specialization degree.
- **Cyberattack simulation:** It indicates if the simulator supports security simulations.

Table 8: Comparison between different IoT simulators

mulator	Scope	Last Update	Language	Type	Layer(s) of IoT Architecture	Evaluated Scale	Mobility	Built-in IoT Standards	Overall Practicality	Target Domain	Cyber Attack Simulation
PWSim 88]	IoT	2016	Java	Event-driven scenarios, resource-constrained environments	Application	Small scale	No	Devices Profile for Web Services (DPWS)	Medium	Generic	No

				and service-oriented [307]						
iFogSim [289]	IoT	2018	Java	Discrete event	Perceptual Network/ Application	Large scale	No	No	Medium	Generic
CloudSim [291]	Data analysis	2016	Java	Discrete event	Application	large scale	Yes	Yes	High	Cloud Analyst
SimIoT [292]	Data analysis	2014	Java	Discrete event	Application	Small scale	No	No	Medium	Generic
IoTSim [297]	Data analysis	2017	Java	MapReduce model	Application	Large scale	No	No	Medium	Generic
Cooja [298]	Network	2018	C /C++	Discrete event	Perceptual Network	Small scale	Yes	Supports all IoT protocols	High	Generic with Focus on power constrained sensors
QualNet [300]	Network	2017	C /C++	Discrete event	Perceptual Network	Large scale	Yes	Zigbee /802.15.4	Medium	Smart city
CupCarbon [304]	Network	2017	Sen Script	Discrete event and agent-based	Perceptual Network	Large scale	Yes	LoRaWAN/ 802.15.4	High	Generic
OMNeT++ [302]	Network	2018	C++	Discrete event	Perceptual Network	Large scale	Yes	Manual extension	Medium	Generic
NS-3 [286]	Network	2018	C++/ Python	Discrete event	Perceptual Network	Large scale	Yes	LoRaWAN 802.15.4 6LoWPAN	High	Generic

8. IoT Applications

The Internet of Things is a modern communication model that envisions a close future, where devices of everyday lif be equipped with transceivers, microcontrollers, sensors, actuators, and appropriate communication protocols that will allow to communicate with each other and with other clients [308] [309] [310] [311]. IoT aims to make the internet immersiv pervasive through enabling easy access and interaction with a wide diversity of IoT devices as surveillance cameras, monit sensors, and home appliances. IoT will promote the development of several applications that utilize the gigantic and diverse ar of data, which is generated by smart devices to provide modern services for companies, citizens and organizations [312] [313

8.1 Sensors in IoT Applications

An IoT network can commonly be described as an area that is occupied by smart sensors, which sense and control th environment [314] [315] [316] [317]. A sensor node is defined from an engineering point of view as an object that con chemical, biological, physical or mechanical parameters into an electrical signal. These sensors are used to measure diff parameters like wind speed (an anemometer), solar radiation or temperature (thermometer), where an IoT application req to include at least one type of sensors to collect data from the IoT environment [314]. Sensor technology is contin improving, accordingly these devices become cheaper, smaller, more energy-efficient, and more intelligent. This will e more applications to be implemented and disseminated such as; environmental monitoring, disaster management, dom human health, public security and early warning systems. Van Laerhoven and Schmidt provided an overview of diverse of sensors that can be utilized in constructing IoT applications. The following section provides a concise preview of sensors:

1. **Light sensor:** It is an electronic device used to detect light. The main function of these sensors is to provide inform about the light density, intensity, type (artificial, sunlight), color temperature (wavelength), and light reflection. Ther many types of light sensors like photodiode, UV-sensors, color sensors, IR sensors, etc. The light sensor is consider be a rich source of data at a very low cost, as it has low energy consumption.
2. **Audio and microphone sensor:** It provides information about various sound types (noise, music, speaking) with minir processing capabilities.
3. **Accelerometer sensor:** It provides information about the motion, the acceleration or the inclination of any mobile dev where angular sensors, accelerometers, and mercury switches are examples of this type of sensors.
4. **Location sensors:** These sensors provide important information about collocation, location, proximity, and positio devices, users or environment. Many applications can be applied using this type of sensors such as GPS, GSM, and ac badge systems [4] [318] [319].
5. **Touch sensors:** Smart devices, which are handled by users, could profit from this type of sensors, as it can be implemen directly with a specific conductive surface, such as skin conductance or indirectly via temperature sensors or light sens These sensors tend to reduce energy consumption significantly, particularly for devices that operate in the user's hand.

6. **Temperature sensors:** These sensors are distinctive as they are easy to use and very cheap. Thus, they can be implemented in many applications such as temperature measurement, fumes and flue gases, body heat detection, and applications of rubber and plastic manufacturing processes, etc.
7. **Pressure sensor:** It is utilized to measure many parameters such as the pressure of liquids or gases, altitude or water level.
8. **Medical Sensors:** Improving the efficiency of biomedical systems and the healthcare infrastructure is one of the most challenging objectives in this era, due to the need of offering quality care to patients with low costs, as well as tackling the shortage problem in nursing staff. IoT sensors can be utilized to resolve the aforementioned issues through monitoring and measuring several medical parameters like blood glucose levels, heart rate, blood pressure, respiration rate, pulse rate and body temperature in the patient's body without any human interference. Medical applications aim to remotely monitor a patient's health and consequently, transfer the sensed data directly to the doctors to take a proper decision [28].
9. **Neural Sensors:** Nowadays, it is easy to comprehend neural signals that come from the human brain, deduce the brain state and train it for a better focus and attention. These operations are known as neuron feedback, while the technology utilized in this operation is called Electroencephalography or also known as a brain-computer interface and totally depends on the electromagnetic field that surrounds humans' brains. This field is generated as a result of the communication between the neurons of the human brain and it is measured in terms of frequencies. Human brain signals can be classified according to their frequencies into gamma, theta, beta, delta, and alpha. Depending on the signal type, it can be concluded whether the brain is wandering in thoughts, calm, etc. in order to train the brain later on to be more focused, have better mental well-being, manage stress and to pay better attention towards things [27] [28].
10. **Environmental and Chemical Sensors:** These sensors are utilized to detect physical, biological, and chemical environmental parameters such as pressure, temperature, humidity, air pollution, and water pollution [320]. A barometer and a thermometer measure the pressure and the temperature parameters, while the air quality is measured through sensors that detect the presence of gases and other polluters in the air. Chemical sensors comprise of transducer and recognition part, where electronic tongue (e-tongue) and electronic nose (e-nose) are examples of applications that are developed depending on this technology [321]. Both of e-nose and e-tongue applications are based on the data generated by chemical sensors, which will be then analyzed by different pattern recognition to identify the stimulus. Furthermore, environmental and chemical sensors play a major role in monitoring the level of pollution in smart city applications [28].
11. **Mobile Phone-Based Sensors:** Today smartphones not only serve as a means of communications and computing operations, but they also provide a valuable set of embedded sensors [322]. These sensors enable the deployment of many applications in various domains, such as accelerometer, camera and microphone, magnetometer, GPS and light sensor.

8.2 Recent IoT Applications

This paradigm finds applications in many distinctive aspects such as medical aids, home automation, mobile healthcare, industrial automation, elderly assistance, smart city, smart grid and many other applications [42]. In this section, some of these applications will be summarized as follows:

8.2.1 Smart Cities

The application of the IoT field toward urban domains is of particular interest. This is coming from a strong motivation of numerous national governments to adopt information and communication technology (ICT) in the management of public affairs, hence realizing the so-called Smart City concept [323]. Smart city aims to make superior utilization of public resources as shown in Figure 16 and to decrease operational costs of public management on many traditional public services such as lighting and parking. Also, it supports the surveillance of public

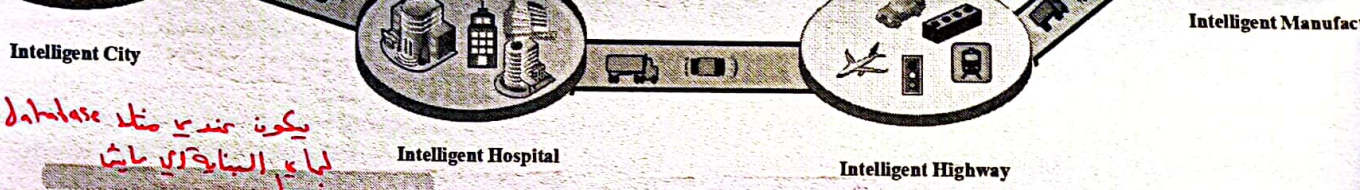


Figure 16: Smart city applications

8.2.1.1 Structural Health of Buildings

This service requires continual monitoring of the specifications of the areas that are prone to the effects of outside ag and the conditions of every building. IoT sensors that are deployed in these buildings should construct a database contai information about the measurement of building structural integrity [324]. There are many types of IoT sensors that can be deplo in this area such as distortion and vibration, which are responsible for measuring buildings stress, atmospheric sensors for sens pollution level of the surrounding area and the sensors that are responsible for measuring the temperature and the humidity of environment [323]. Employing IoT technology in this field reduces the cost of human periodic checking on building health thro deploying a number of wireless sensors on the building and the surrounding area.

8.2.1.2 Waste Management

Waste disposal is an essential problem in many modern cities, because of both the storage constraints of garbage in landf and the cost of this service. Applying IoT in this domain will lead to significant ecological advantages and significant cost savi For example, the utilization of smart garbage collection to detect waste level and to optimize the garbage truck route will decre the cost of the garbage collection process and will enhance the quality of recycling. To attain these objectives, IoT must connect smart garbage collectors with a control center that processes the sensed data by an optimization software and then determine best management of this operation [323].

8.2.1.3 Air Quality and Noise Monitoring

Sound and air pollution are escalating problems nowadays. It is important to monitor air quality and keep it with acceptable limits for a healthy living and a better future for all organisms. Air quality monitoring gives estimations of gases a toxic concentrations to be then analyzed and interpreted, allowing authorities to monitor air pollution in distinctive zon consequently taking action against any pollution. In such a way, individuals can find the healthiest places to practice outdo exercises, also they can access their favoured training applications that are connected to IoT infrastructure [325] [326].

8.2.1.4 Traffic Congestion

Traffic management is an issue that most cities confront today. Investing in smart traffic solutions makes sense, as m than half of the world's population were reported living in cities in 2012 [28]. Hence, many cities try to improve transportation deploying smart services like smart traffic signals and developing applications for smart parking. Furthermore, improving s transportation systems will increase transportation capacity and make traveling safer, efficient, and secure [327].
Embedding IoT sensors in smart traffic areas will alleviate congestion, respond rapidly to any accident or incident, and manage daily traffic in smart transportation environments. The major objectives of smart transportation systems are to minimize tra congestion and provide the individual with hassle-free and easy parking. Furthermore, it will help to avoid accidents by prop routing the traffic and informing the drivers about other bad drivers [28]. Sensors technologies that control these applications

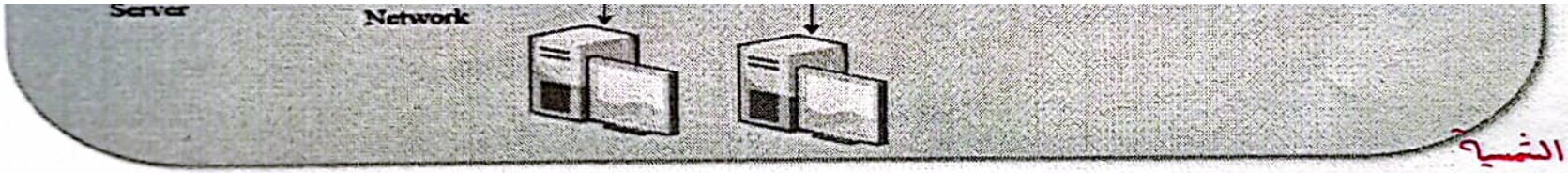


Figure 17: PGI system architecture

* زيد موضوع الطاقة الشمسية
انه يكون عندي electrical network وتتوزع ال power
بشكل صحيح زيد احط sensor واعطي لكل واحد حب الطاقة او عديه دافع .

8.2.1 5 Smart Grid

It is an electrical network that smartly connects and integrates the activities of many users, whether they are producers or consumers or those that do both, to effectively afford economic, sustainable and secure power resources. A Smart Grid employs inventive services and products with intelligent surveillance, self-healing technologies, communication, and control to accomplish the following purposes [334]:

1. Permitting customers to optimize the operations of the smart grid system.

2. Providing customers with more data and choices of
3. Significantly diminishing the environmental effect on the entire power supply
4. Providing enhanced levels of security and reliability on power supply systems.
5. Enabling distribution of the generation and utilization processes of renewable energy resources.

8.2.1.6 Automation and Salubrity of Public Buildings

This significant application aims to achieve salubrity of the environment and to alleviate energy consumption problem i public buildings such as museums, administration offices and schools [323], which improves the level of comfort for the individual enhances the efficiency, while it decreases the costs of heating and cooling [335]. This is accomplished by utilizing appropria types of actuators and sensors that control humidity, temperature, and lights.

8.2.1.7 Smart Water Systems

It develops a modern approach that promotes water security from significant future risks such as rapid urbanizatio population growth, weak policies, aging infrastructure and climate changes, where these factors will increase the burdens on wat resources. Water is delivered to consumers through complex distribution systems. Thus, these systems should supply potable a safe water with adequate pressure. Nevertheless, any failure that infects these systems will lead to waste and declination of t quality of water. Hence, a novel water management procedure is robustly required to carefully control water distribution netwo and to detect any deficiencies promptly. The primary objectives of Smart Water Networks (SWNS) are to construct a compl surveillance system, data acquisition, integrating sensors technology, securing the gathered information, information analysis, a take decisions in real-time [336]. SWN operation comprises of many steps, where the first step is to have a schematic visualizati to collect full information of water network, like pipes, tanks, air valves, pumps, and stabilizers, in order to group them in the ne step in geographic information framework. After that, a set of sensors will be deployed to continuously sense many water paramet such as pressure, quality, and flow. Finally, the sensed data will be transmitted through communication channels to be analyzed an information system to take a suitable action [337].

8.2.2 Medical and Healthcare Applications

Wireless body area networks and WSNs that are utilized in both healthcare and medical applications have received important interest, as they have major roles in remote monitoring of a patient's situation in real-time, life quality enhancement the elderly via smart environment, drugs and medical database administrator, avoidance of critical patient situations, welf services, etc. According to that, it is clear that applying IoT in medical applications will improve radically medical environme [313]. For example, smart health applications allow elderly and patients who are suffering from serious health conditions to l independently apart from hospital restrictions, through utilizing IoT sensors, which continually monitor and record differ parameters of their health conditions. Subsequently, delivering warnings in case of finding any unusual indicator. Smart senso which are dedicated, for healthcare can measure, monitor and analyze different health status conditions such as heart rate, blo pressure, oxygen saturation in the blood and glucose levels. After measuring the aforementioned parameters, the sensed data w be transmitted to a specific database in order to be analyzed and accordingly to take a proper action, which will enhance the patien health as shown in Figure 18 [28] [313].

Stress recognition is another healthcare application that is based on sensors of smartphones, which sense the stress level an individual. This can be achieved through measuring physiological and behavioral data such as blood pressure, skin conductan heart rate, pupil diameter, and cortisol level to identify whether the person is feeling stress or not [338].

8.2.3 Agricultural Applications

Agriculture plays an important role in any country's economy as it provides extensive employment opportunities individuals. However, numerous factors affect this field such as soil moisture, carbon dioxide and changes in temperature, wh affect the crops, yield. Thus, it is vital to have surveillance systems on these factors to manage harvest growth and to raise agricultu production yield by deploying IoT sensors in agricultural areas [339]. These sensors are able to monitor different environme parameters such as humidity, temperature, barometric pressure, and luminosity. Any agricultural smart application comprises two sides, the transmitter side and the gateway receiver side. The transmitter side consists of many sensors that are connected t wireless network in order to sense different agricultural parameters, while the receiver side monitors and analyzes the sens parameters, which will be displayed by a user through a web interface as shown in Figure 19.

8.2.4 Smart Home (SH)

SH technology has changed individual life by providing connectivity between everyone and everything regardless of t place and the time. This application changes a traditional home into an automated building with installed and controlled sm devices such as heating, air conditioning, ventilation, security systems and lightings as shown in Figure 20. These systems, wh consist of sensors and switches that are sometimes called gateways, communicate with a central station that can be control through a user interface installed in a mobile phone, tablet or computer and managed by IoT technology [340]. Smart home syst aims to improve domestic comfort, security, leisure, and convenience, while minifying energy consumption through optimiz domestic energy management techniques [341]. SH applications are characterized by the following features:

1. Compatibility with distinctive communication protocols: It can merge numerous heterogeneous communicati techniques through installing different communication interfaces on a home gateway.
2. Widespread services: With the utilization of widespread access networks, real-time smart home data can be obtai easily regardless of where the clients are.
3. Comprehensive perception: Real-time surveillance of domestic and comprehensive perception can be attained deploying an assortment of physical and logical sensors.

4. Easily to be controlled: Since SH applications can be managed via mobile phones, PCs and other communication devices.

+ هو عملية الصناعة تحليلها Smart عن طريق اد Cost, quality, delivery وانها اخفف من التدخل البشري بقلل التكلفة

8.2.5 Smart Manufacturing System (SMS)

Maintainable manufacturing competitiveness relies on its capabilities with respect to quality, cost, delivery, and flexibility [342]. SMS tries to maximize those capabilities through utilizing advanced technologies, which promote quick flow and widespread utilization of digital data inside and among manufacturing systems. Also, it integrates information and communication technologies with smart software applications to:

1. Enhance the utilization of material, energy, and labor to produce high quality and customized items to be delivered on time.
2. Rapidly respond to changes in supply chains and mart demands.

Smart manufacturing model is distinctive from other manufacturing paradigms as it determines a vision of the next manufacturing generation with improved capabilities [343]. SMS adapts to any new circumstances by utilizing real-time information for intelligent decision-making and by predicting and preventing any failure proactively.

8.2.6 Internet of Robotics Things (IoRT)

In diverse industries or even in offices or homes, robotics come in all sizes and shapes from greeting robotics in restaurants, retail stores or hotels to heavy robot arms in factories. The internet of robotics is an emerging technology that integrates robots as an object into an IoT environment to enable connections through different protocols. IoRT integrates smart robots through the internet to perform personal activities or different professional operations as monitoring activities and events, controlling objects in the real-world and manufacturing. In IoRT application, multiple intelligent sensors and smaller robots are connected and collaborated in an orchestrated manner to achieve the goal of large robotic. There are many applications that are implemented through IoRT such as a self-driving vehicle, software robots to avoid human errors and save time, Smart Manufacturing (Industry 4.0), adaptive digital factory and automated IT processing applications

بصير عندي شبكة من الروبوتات مع بعض وتنسيق بينهم اكتر اشي بالصانع مع اد Smart Manu...

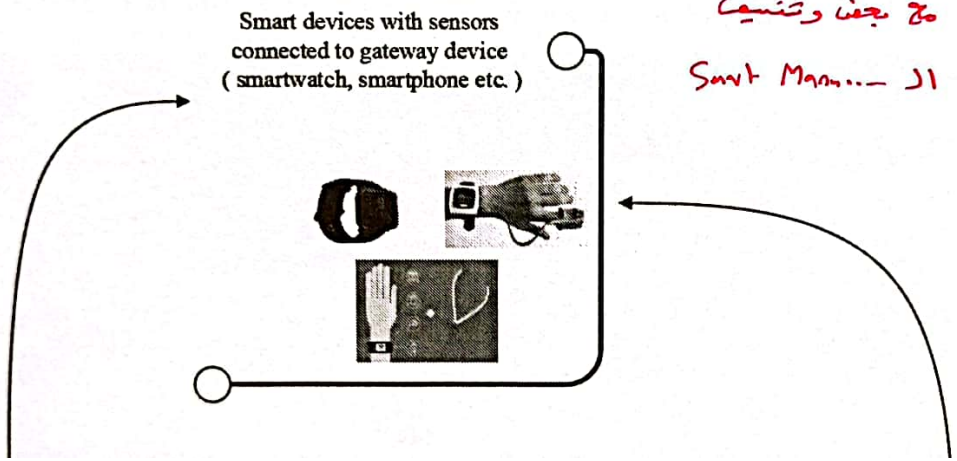


Figure 20: Applications of smart home technology [344]

8.2.7 Oil and Gas

IoT paradigm has found its way through the oil and gas domain. As of now, many IoT companies help factory mana field staff and machine operators to improve production, protect the safety of employees in work environments and predict the at which machines require maintenance. IoT technology permits machines, devices, and equipment to collaborate and commu with each other, which will enable oil and gas companies to create applications, manage and store data and utilize suitable se protocols based on scientific methodologies.

9. Broad and Open Research Challenges

IoT is a fabulous technology concept that for a long period was merely a dream. Nowadays, IoT has taken the world storm and it is expanding with an unbelievable rate. Morgan Stanley predicted that the number of smart and heterogeneous that will be connected to the internet would exceed 75 billion devices in 2020 [41] [345]. However, IoT services, applicatio devices face numerous issues and challenges that are deemed to be a primary hindrance in the implementation of IoT from d

* حنشتغل IoT infrastructor و Wireless
↳ حلينا الـ Delay وهلد صعبنا الـ Power .
* اكبر 2 challenges للـ IoT هم: Power and Delay *
* هيك حنشرح عن الـ IoT Infrastructure .

Chapter 3

كيفية التعامل مع موضوع الـ energy

The LEACH Protocol Architecture

↳ Fully distributed (الـ based-station ما الو علاقة من قريب او بعيد)

Wireless microsensor networks will enable reliable monitoring of remote areas. These networks are essentially data-gathering networks where the data are highly correlated and the end-user requires a high-level description of the environment the nodes are sensing. In addition, these networks require ease of deployment, long system lifetime, and low-latency data transfers. This is a very different paradigm than traditional wireless networks that require point-to-point connectivity, have uncorrelated data, and often can rely on a fixed infrastructure. The limited battery capacity of microsensor nodes and the large amount of data that each node may produce translates to the need for high application-perceived performance at a minimum cost, in terms of energy and latency. A cross-layer or application-specific protocol architecture can meet these specifications by exposing lower layers of the protocol stack to the requirements of the application.

* مثال بـ CH4 ← slide84

To meet the requirements of wireless microsensor networks, we developed LEACH (Low-Energy Adaptive Clustering Hierarchy), an application-specific protocol architecture (see Figure 3-1) [40]. LEACH is a clustering-based protocol that includes the following features:

* شرح بسيط للـ energy model بالـ Sensor

فكرته انه اعين Cluster head للـ nodes هي اللي بتجمع الـ data وبتبعتها للـ base Station .

- randomized, adaptive, self-configuring cluster formation,
- localized control for data transfers,
- low-energy media access, and
- application-specific data processing, such as data aggregation.

* كل الـ Nodes بكونوا نفس بعض فمين اي نختاره cluster-head منعرف لو اخترنا واحد بفضل هو دايمًا الـ cluster-head مشكلة لانه حتخلص الـ energy تبعته بأقرب وقت فلازم نعمل طريقة ليصير في Rotation حال الـ nodes لاختيار الـ cluster-head

The application that typical microsensor networks support is the remote monitoring of an environment. Since individual nodes' data are correlated in a microsensor network, the end-user does not require all the (redundant) data; rather, the end-user needs a high-level function of the data

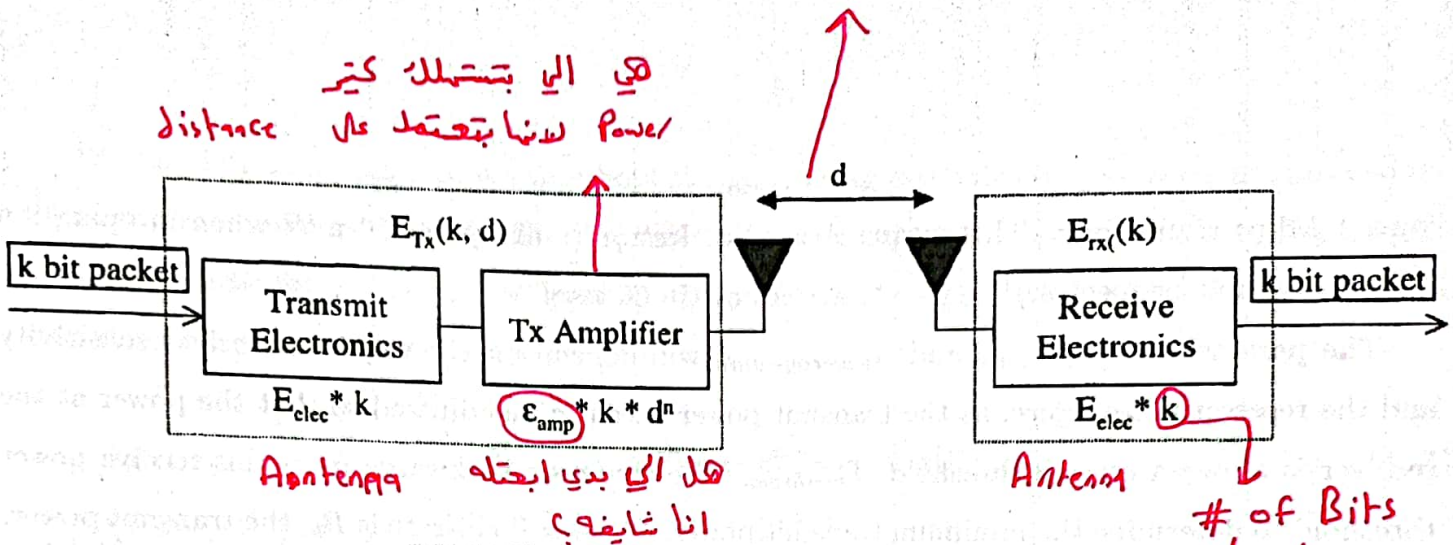


Figure 4-1: Radio energy dissipation model.

the power attenuation is dependent on the distance between the transmitter and receiver. For relatively short distances, the propagation loss can be modeled as inversely proportional to d^2, whereas for longer distances, the propagation loss can be modeled as inversely proportional to d^4. Power control can be used to invert this loss by setting the power amplifier to ensure a certain power at the receiver. Thus, to transmit an l-bit message a distance d, the radio expends:

$$E_{Tx}(l, d) = E_{Tx-elec}(l) + E_{Tx-amp}(l, d) \tag{4.5}$$

$$E_{Tx}(l, d) = \begin{cases} lE_{elec} + l\epsilon_{friss-amp}d^2 & : \quad d < d_{crossover} \\ lE_{elec} + l\epsilon_{two-ray-amp}d^4 & : \quad d \geq d_{crossover} \end{cases} \tag{4.6}$$

and to receive this message, the radio expends:

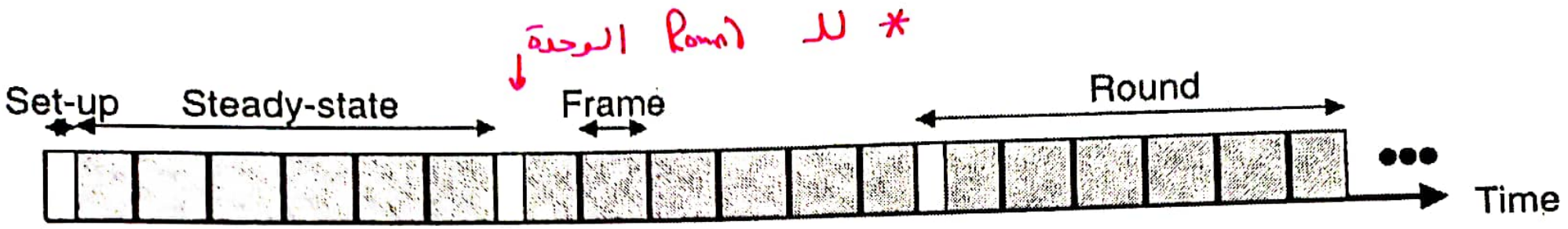


Figure 3-2: Time-line showing LEACH operation. Adaptive clusters are formed during the set-up phase and data transfers occur during the steady-state phase.

Thus, when a cluster-head node dies (e.g., uses up all its battery energy), all the nodes that belong to the cluster lose communication ability. Thus LEACH incorporates randomized rotation of the

announcment هو الي بيعت ال cluster - head * ال
Transmit للباقي انو cluster - head عشان يتجنب حد غيره
Power يبعت عشان ما يصير فيه interference ولازم تكون ال
عنده عالية ليوصل الرسالة للكل.

3.1.1 Determining Cluster-Head Nodes

Given that we want to produce clusters using a distributed protocol, what are the important goals that we are trying to achieve? What constitutes a good cluster formation? To begin with, we want to design the algorithm such that there are a certain number of clusters, k, during each round. Second, we want to try to evenly distribute the energy dissipation among all the nodes in the network so that there are no overly-utilized nodes that will run out of energy before the others. This will maximize the time until the first node death. As being a cluster-head node is much more energy-intensive than being a non-cluster-head node (since the cluster-head node must receive data from all the nodes in the cluster, perform signal processing functions on the data, and transmit the data to an end-user who may be far away), evenly distributing the energy load among all the nodes in the network requires that each node take its turn as cluster-head. Therefore, the cluster formation algorithm should be designed such that nodes are cluster-heads approximately the same amount of time, assuming all the nodes start with the same amount of energy. Finally, we would like the cluster-head nodes to be spread throughout the network, as this will minimize the distance the non-cluster-head nodes need to send their data.

In LEACH, sensors elect themselves to be cluster-heads at the beginning of round $r+1$ (which starts at time t) with a certain probability, $P_i(t)$. This probability is chosen such that the expected number of cluster-head nodes for this round is k. Thus:

$$E[\#\text{ CH}] = \sum_{i=1}^{N} P_i(t) * 1 = k \tag{3.1}$$

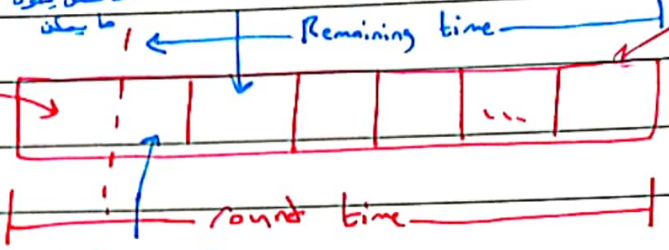
where N is the total number of nodes in the network. Ensuring that all nodes are cluster-heads the same number of times requires each node to be a cluster-head once in $\frac{N}{k}$ rounds. Combining these constraints gives the following probability for each node i to be a cluster-head at time t:

$$P_i(t) = \begin{cases} \frac{k}{N - k*(r \bmod \frac{N}{k})} & : \quad C_i(t) = 1 \\ 0 & : \quad C_i(t) = 0 \end{cases} \tag{3.2}$$

where r is the number of rounds that have passed and $C_i(t) = 0$ if node i has already been a cluster-head in the most recent ($r \bmod \frac{N}{k}$) rounds and 1 otherwise. Therefore, only nodes that have not already been cluster-heads recently, and which presumably have more energy available than nodes that have recently performed this energy-intensive function, may become cluster-heads

Round كل يتحدد Frame ال طول
لازم فيه node ممكن يروحوا
الأفضل يكون عدد الـ Round أكبر ما يمكن

Ch J



Clustering
اسمه بين الـ head
وبين كل Cluster
(overhead)

لكل node
انه متى كل node
جه يبعت عتان ما يتداخلوا
في بعض ولما يكون شغيل
يدخل بالـ sleep هيك
بوفر energy لانه لو كان
شغال ممكن يصير يستقبل طلبات لو
صار شغيل بنخدم energy

Time division multiple Access (TDMA)
↳ To avoid interference between
cluster's member
بقسم الـ Time اللي ضايل على عدد الـ Sensors اللي عندي
بالشبكة no. Sensors +1

Slot time = Remaining time / # of Sensors for +1 Cluster

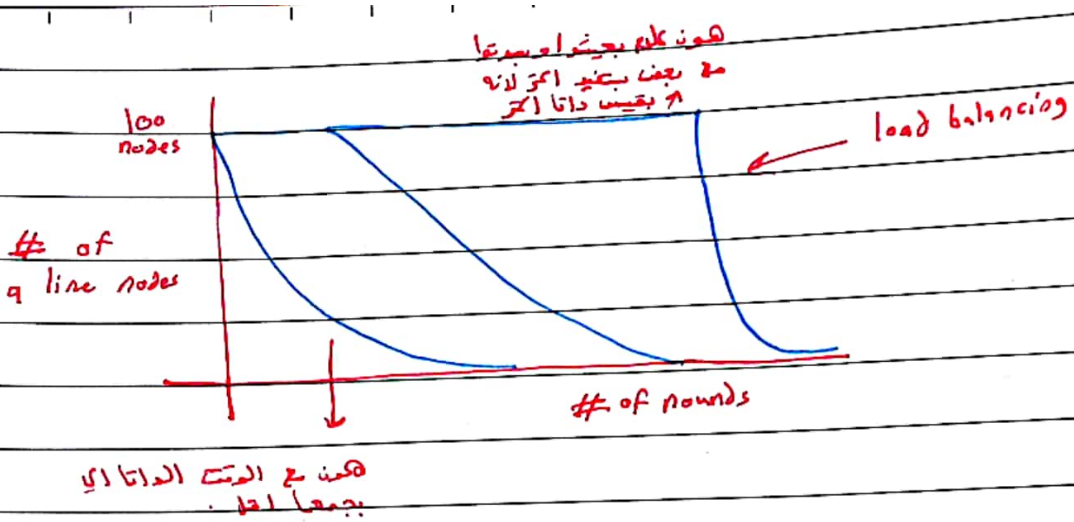
* كل Sensor بيعت بالـ Slot تابعه.

عتان ابعت للـ base station لانه لازم بعد كل ابعتلك Round time

- Note: LEACH is a fully distributed algorithim.

* كل node عنده معادلة انه قديه الـ Probab. انه هو يكون الـ Cluster head بين 0 و 1
اذا طلعت الـ Prob اكبر من الـ Random number generator انا بطلع انا Cluster head
الحالة بتعتمد على كمية اشياء تيجي مثلا هل انا كنت قبلها cluster head ولا لا

Target: is to maximize the number of rounds which reflect the network lifetime.



Slide 59:

$$P_i(t) = \begin{cases} \dfrac{K}{N - K\left(r \bmod \frac{N}{K}\right)}, & C_i(t) = 1 \\ 0, & C_i(t) = 0 \end{cases}$$

- K → expected # of clusters in the network
- N → # of nodes in the network
- r → round #
- $r \bmod \frac{N}{K}$ → Batch Reset
- $P_i(t)$ → Prob. that a node i is a cluster head
- $C_i(t) = 1$ → The node i is eligible
- $C_i(t) = 0$ → Node is not eligible to be a cluster head (Served)

*Tip: I need to reset the calculations every batch ($\frac{N}{K} = 5$ rounds) | Batch = 5 rounds |

Note: After $\frac{N}{K}$ rounds, all nodes should have served as CHs (CH ↓ cluster head)

$N = 10$, $K = 2$ → Expected # of clusters in the network = 2 → هون عندي 2 clusters

بعد 5 round لازم يكونوا كلهم خدموا CH.

* لازم أعمل Setup كل Round عشان أتأكد إنه كل ال Nodes ← Alive. (Power)
صح هي Overhead بس مهمة

ال Prob لازم تقل بزيادة ال rounds

* كل $\frac{N}{K}$ ← rounds بعمل Reset عن طريق ال $r \bmod \frac{N}{K}$ عشان بعد ال $\frac{N}{K}$ يرجع من أول

Nodes per cluster $= \frac{10}{2} = 5$ nodes

* أول Round احتمال اي اشي يكون CH ← $\frac{K}{N}$

0.2 0.2 0.2 0.2 0.2 ✓

O O O O O

$\frac{1}{5}$ $\frac{1}{5}$ $\frac{1}{5}$ $\frac{1}{5}$ $\frac{1}{5}$ ✓ (Round 1: r_0) → every cluster has 5 nodes

O O O O O

! R_0

* ثاني Round الي ضل nodes ضلوا بتصير 0.25 $(\frac{1}{4})$ ،

Target: during the five rounds, all 5 nodes should have served as CHs.
↳ why? load balancing

R_1 : 0.25 0.25 0.25 0.25 0 — 0.25 0.25 0.25 0.25 0

Note: ① Every node will generate a random # between 0 & 1
② If $P_i(t) \le$ RNG, then this node (node i)
~~# $P_i(t) \ge$~~ will not declare itself as a CH/Utilizing
~~Serial Sense multiple Access (CSMA)~~
Otherwise, it will announce itself (broadcast a message to all nodes in the network) Utilizing CSMA MAC Protocol

CSMA: Carrier Sense multiple Access

* اذا واحد بعث وراها واحد تاني يبعث بضل يستنى من CSMA .

* اما ال node للي CH بدي انضم الو كل حسب مين ال Signal الاقوى

R_2 : $\frac{1}{3}$ $\frac{1}{3}$ $\frac{1}{3}$ 0 0 — $\frac{1}{3}$ $\frac{1}{3}$ $\frac{1}{3}$ 0 0

R_3 : $\frac{1}{2}$ $\frac{1}{2}$ 0 0 0 — $\frac{1}{2}$ $\frac{1}{2}$ 0 0 0

R_4 : 1 0 0 0 0 — 1 0 0 0 0

* عند R_5 ← new batch يعني بنعملها Reset من اول.
* اذا ما رحلا معرفة كام منا CH خلص بجعلها برجع للـ base station .
* ممكن بسبب ال RNG كتير مرات ما بصير في CH .

Slide 5 : من ضمن الـ Apps اللي ممكن أعملهم إني أعمل deployment لمجموعة من الـ Sensors لتزودني بمعلومات، نعتبر إنو إنا الـ Base-Station وهو device بعمل Collecting لكل الـ data من الـ Sensors.

* الـ Sensors بتكون devices low power, low cost, low Capability.

* كيف بدي أتعامل مع موضوع الـ energy إنا Base-Stations؟ لو كل Sensor بده يبعت اشي لازم Transmission Power تكون عالية عشان توصل الـ nodes اللي جنبه ولو ما كانت عالية ممكن ما توصل لـ Base-Station.

* لو كل Sensor بده يبعت دايركت لـ Base-Station حتستهلك Power عالي وبصير interference بين الـ Nodes وتداخل حيمكن وصول المعلومة لـ Nodes معينة أكتر من مرة.

* الـ Base-Station بنفرض إنو الـ Power تبعو عالي لأنو الـ Capability High و High Power.

* عشان ننظم هذا الموضوع أكتر فيه أكتر من طريقة منها الـ LEACH Protocol اللي بحلل يعالج الـ Power Consumption وشغلة إرسال الـ nodes لـ Base اللي بتستهلك Power عالية بسبب احتياجها لـ Transmission Power عالية.

* بالمفهوم الطبيعي كان عندي كل node بعمل sense لـ data وبعتها لـ Base-Station بس هون الفكرة إنو اعتبر الـ nodes لمجموعات كل مجموعة فيها Cluster head هو اللي بجمع الداتا وبعتها لـ Base-Station.

*Note! Receive power not depend on spaces like transmit power.

* كل الـ nodes بكونوا homogeneous نفس الـ initial وبنفس الـ Communication Protocol

* كيف الـ node بده يقرر إنو هو Cluster-head؟ مثلاً" الـ Base station بعدد الـ Round time (مثلاً كل ما حكينا لازم الشبكة تزودني بالـ data اللي Sensing Tables) داخل الـ Round time في قسمين:

- Steady State/ Setup ← بصير فيه transmit من الـ node لـ Cluster head
  - بصير فيه Clustering (overhead) بتعين كل Cluster head وبين الـ Cluster members.
  - لـ Base Station.

* طيب هنا رح نحدد انه مين الـ cluster-head بين هدول الـ nodes الي بالـ cluster:

*Note: LEACH is a fully distributed Algorithm

* صار كل node عنده معادلة الي هي P(being a cluster head) بحسبها وبطلع Random no. generator بين الـ 0 و 1 (بين 0 و 1 لانه المعادلة عبارة عن Prob.) لو طلعت النتيجة انه مثلا 0.7 = cluster head معناها عالية فبتطلع حال Random number، لو كان الـ Random no. الي طلع عندي اقل من الـ Probability معناته انا Cluster-Head وبعث announcement انه انا cluster head ولو اكبر معناه انا مش cluster head.

* هاي المعادلة بتعتمد على نقطة مهمة انه هل انا خدمت كـ cluster head قبل هيك اولا لو خدمت قبل هيك احتماليتي اني اكون cluster head وهاد منطقي عشان Rotation يصير بين الـ nodes.

* كل Round بصير عندي Setup (Clustering) لانه في nodes بتكون تموت (تخلص الـ energy تاعتها).

* الهدف من هاد كله انه تكون الـ Rounds اكبر ما يمكن وهذا اللي بيسمى Network lifetime.

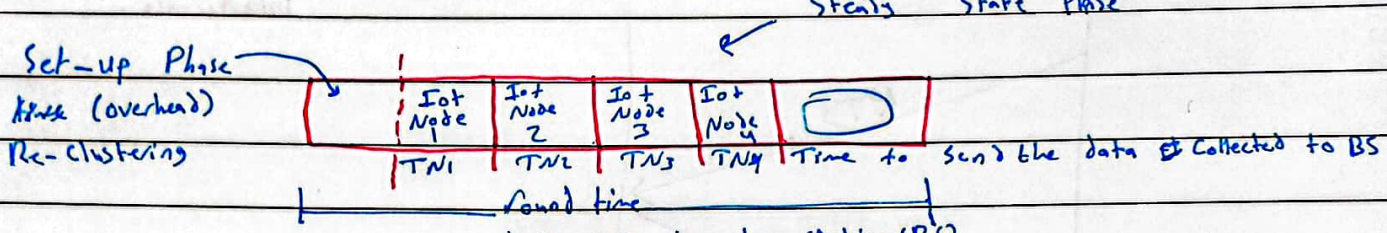
~~*~~

* Target: is to maximize the number of rounds which reflects the network lifetime.

Interference avoidance MAC Protocols:

① CSMA / CA
↳ Carrier Sense Multiple Access / Collision avoidance.

② TDMA: Time division Multiple Access
↳ Time is slotted



CH main task is to split the remaining time of the round over its member. Let us assume that its members are 4

Tip: There are two important Circuits in any IoT node/Devise, namely Sensing Circuit & Transmit/Receive Circuit.

* TDMA يختلف عن طريقة CSMA ان: Interference

③ FDMA / Frequency division Multiple Access
↳ Every Signal/Message will be transmitted Using different Carrier frequency.

Big Problem: Spectrum limitation

④ CDMA: Code division multiple Access.
Spreading Codes will be used during transmissions.

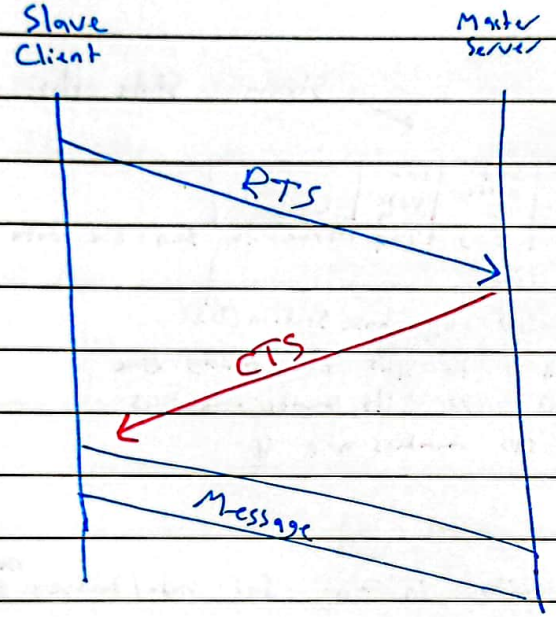
Tip: Each node assigned a unique Spreading Code to be used during transmissions.

limitation: a Bank of Spreading Codes Should be employed at Rx transmission (Memory + Delay)

Five Appl

حسب الـ code بس يوصل بيعرف لازم هذا الـ node code ، بس مشكلة بتتاج لـ Memory كبيرة ليستوعب كل الـ nodes وهذا overhead

⑤ RTS / CTS
↳ Request to send → Clear to send



* Handshaking to avoid interference.

* Will not work in IoT clustering? why? CH is not known in the early stages.

at round $r+1$. The expected number of nodes that have not been cluster-heads in the first r rounds is $N - k*r$. After $\frac{N}{k}$ rounds, all nodes are expected to have been cluster-head once, following which they are all eligible to perform this task in the next sequence of rounds. Since $C_i(t)$ is 1 if node i is eligible to be a cluster-head at time t and 0 otherwise, the term $\sum_{i=1}^{N} C_i(t)$ represents the total number of nodes that are eligible to be a cluster-head at time t, and

$$E[\sum_{i=1}^{N} C_i(t)] = N - k * (r \bmod \frac{N}{k}) \tag{3.3}$$

This ensures that the energy at each node is approximately equal after every $\frac{N}{k}$ rounds. Using Equations 3.2 and 3.3, the expected number of cluster-heads per round is:

$$\begin{aligned} E[\#\ \text{CH}] &= \sum_{i=1}^{N} P_i(t) * 1 \\ &= (N - k * (r \bmod \frac{N}{k})) * \frac{k}{N - k * (r \bmod \frac{N}{k})} \\ &= k \end{aligned} \tag{3.4}$$

The optimal k can be determined analytically based on the energy dissipation models for computation and communication and the network topology. Analysis and simulation to determine the optimal k will be described in detail in Section 4.3.

This choice of probability for becoming a cluster-head is based on the assumption that all nodes start with an equal amount of energy, and that all nodes have data to send during each frame. If nodes begin with different amounts of energy (or an event-driven model is used, whereby nodes only send data when some event occurs in the environment), the nodes with more energy should be cluster-heads more often than the nodes with less energy, in order to ensure that all nodes die at approximately the same time. This can be achieved by setting the probability of becoming a cluster-head as a function of a node's energy level relative to the aggregate energy remaining in the network, rather than purely as a function of the number of times the node has been cluster-head:

$$P_i(t) = \frac{E_i(t)}{E_{total}(t)} k \tag{3.5}$$

where $E_i(t)$ is the current energy of node i, and

$$E_{total}(t) = \sum_{i=1}^{N} E_i(t) \tag{3.6}$$

Using these probabilities, the nodes with higher energy are more likely to become cluster-heads than nodes with less energy. The expected number of cluster-head nodes is:

$$\begin{aligned} E[\#\ \text{CH}] &= \sum_{i=1}^{N} P_i(t) * 1 \\ &= (\frac{E_1(t)}{E_{total}} + \ldots + \frac{E_N(t)}{E_{total}})k \\ &= k \end{aligned} \tag{3.7}$$

Equation 3.2 can be approximated by Equation 3.5 when the nodes begin with equal energy, E_o. If a node has been a cluster-head in the last $r < \frac{N}{k}$ rounds, its energy is approximately $E_o - E_{CH}$, where E_{CH} is a large number less than E_o. If the node has not been a cluster-head in the last r rounds, its energy is approximately E_o, since being a non-cluster-head node does not require much energy from the node relative to being a cluster-head node. Since it is expected that kr nodes have been cluster-heads and $N - kr$ nodes have not been cluster-heads in the last r rounds, the total expected energy is given by:

$$E[E_{total}] = E_o(N - kr) + (E_o - E_{CH})(kr) \tag{3.8}$$

Therefore, Equation 3.5 becomes:

$$E[P_i(t)] = \begin{cases} \frac{E_o k}{E_o(N-kr)+(E_o-E_{CH})kr} & : C_i(t) = 1 \\ \frac{(E_o - E_{CH})k}{E_o(N-kr)+(E_o-E_{CH})kr} & : C_i(t) = 0 \end{cases} \tag{3.9}$$

Since $E_o >> (E_o - E_{CH})$, this can be simplified to:

$$E[P_i(t)] \approx \begin{cases} \frac{k}{N-kr} & : C_i(t) = 1 \\ 0 & : C_i(t) = 0 \end{cases} \tag{3.10}$$

Thus the expected probability for each node becoming a cluster-head at round t is exactly the same as in Equation 3.2 (for $r < \frac{N}{k}$).

Using the probabilities in Equation 3.5 requires that each node have an estimate of the total energy of all the nodes in the network. This requires a routing protocol that allows each node to determine the total energy, whereas the probabilities in Equation 3.2 enable each node to make completely autonomous decisions. One approach to avoid the routing protocol might be to approximate the aggregate network energy by averaging the energy of the nodes in each cluster and multiplying by N.

3.1.2 Set-up Phase

Once the nodes have elected themselves to be cluster-heads using the probabilities in Equation 3.2 or 3.5, the cluster-head nodes must let all the other nodes in the network know that they have chosen this role for the current round. To do this, each cluster-head node broadcasts an advertisement message (ADV) using a non-persistent carrier-sense multiple access (CSMA) MAC protocol [69]. This message is a small message containing the node's ID and a header that distinguishes this message as an announcement message. However, this message must be broadcast to reach all of the nodes in the network. There are two reasons for this. First, ensuring that all nodes hear the advertisement essentially eliminates collisions when CSMA is used, since there is no hidden terminal problem (as discussed in Section 2.1.2). Second, since there is no guarantee that the nodes that elect themselves to be cluster-heads are spread evenly throughout the network, using enough power to reach all nodes ensures that every node can become part of a cluster. If the power of the advertisement messages was reduced, some nodes on the edge of the network may not receive any announcements and therefore may not be able to participate in this round of the protocol. Furthermore, since these advertisement messages are small, the increased power to reach all nodes in the network is not a burden. Therefore, the transmit power is set high enough that all nodes within the network can hear the advertisement message.

Each non-cluster-head node determines to which cluster it belongs by choosing the cluster-head that requires the minimum communication energy, based on the *received signal strength* of the advertisement from each cluster-head. Assuming symmetric propagation channels for pure signal strength[1], the cluster-head advertisement heard with the largest signal strength is the cluster-head to whom the minimum amount of transmitted energy is needed for communication. Note that

[1]In the absence of mobile objects moving into or out of the channel, the pure signal strength attenuation of a message sent from a transmitter to a receiver will be the same as the attenuation of a message sent from the receiver to the transmitter because the electromagnetic wave traverses the same path in both cases.

typically this will be the cluster-head closest to the sensor. However, if there is some obstacle impeding the communication between two physically close nodes (e.g., a building, a tree, etc.) such that communication with another cluster-head, located further away, is easier, the sensor will choose the cluster-head that is spatially further away but "closer" in a communication sense. In the case of ties, a random cluster-head is chosen.

After each node has decided to which cluster it belongs, it must inform the cluster-head node that it will be a member of the cluster. Each node transmits a join-request message (Join-REQ) back to the chosen cluster-head using a non-persistent CSMA MAC protocol. This message is again a short message, consisting of the node's ID, the cluster-head's ID, and a header. Since the node has an idea of the relative power needed to reach the cluster-head (based on the received power of the advertisement message), it could adjust its transmit power to this level. However, this approach suffers from the hidden-terminal problem; if a node close to the cluster-head is currently transmitting a join-request message using low transmit power, the remaining nodes in the cluster cannot sense that this transmission is occurring and may decide to transmit their own join-request messages. Since these messages are small, it is more energy-efficient to just increase the transmit power of the join-request messages than to use an 802.11 approach of transmitting request-to-send and clear-to-send (RTS-CTS) messages [24]. This is because the cluster-head does not know the size of its cluster and would need to transmit the CTS message using large power to reach all potential cluster members. In addition, simply increasing the transmit power reduces the latency and increases the sleep time allowed for all the nodes compared to an RTS-CTS approach. Therefore, the nodes use a large power for transmissions of the short join-request messages to the cluster-heads.

The cluster-heads in LEACH act as local control centers to coordinate the data transmissions in their cluster. The cluster-head node sets up a TDMA schedule and transmits this schedule to the nodes in the cluster. This ensures that there are no collisions among data messages and also allows the radio components of each non-cluster-head node to be turned off at all times except during their transmit time, thus minimizing the energy dissipated by the individual sensors. After the TDMA schedule is known by all nodes in the cluster, the set-up phase is complete and the steady-state operation (data transmission) can begin.

A flow-graph of this distributed cluster formation algorithm is shown in Figure 3-3. Figure 3-4 shows an example of the clusters formed during two different rounds of LEACH.

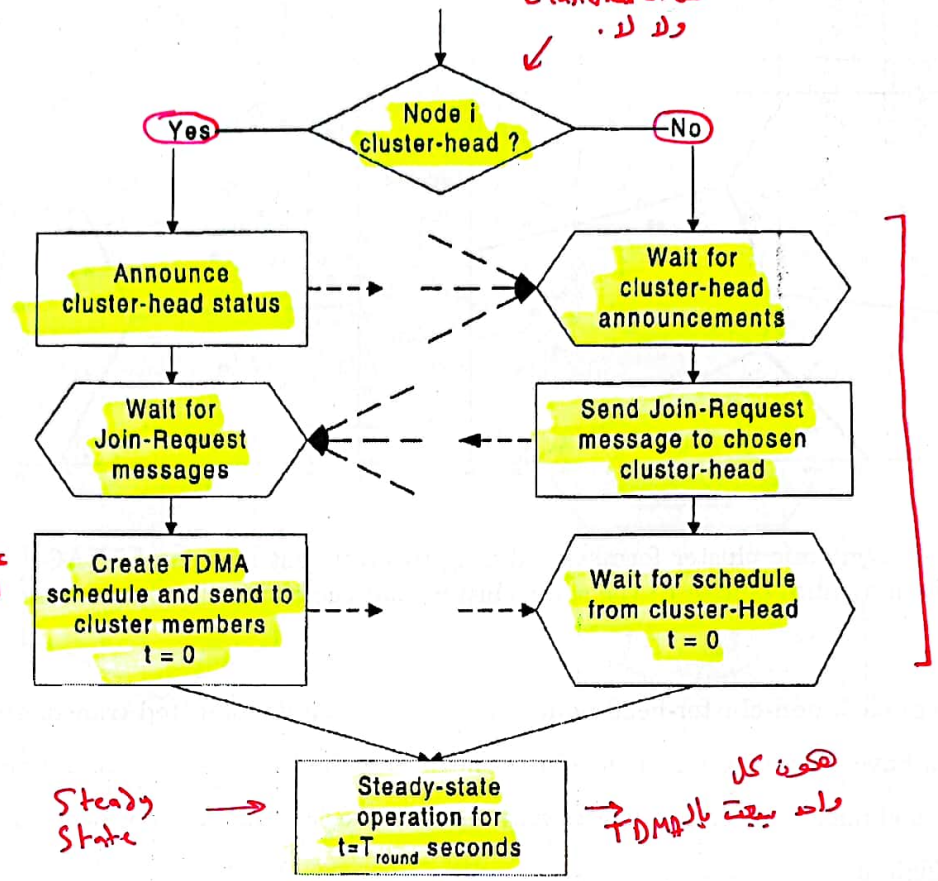


Figure 3-3: Flow-graph of the distributed cluster formation algorithm for LEACH.

3.2 Steady-state Phase

The steady-state operation is broken into frames (see Figure 3-2), where nodes send their data to the cluster-head at most once per frame during their allocated transmission slot. Each slot in which a node transmits data is constant, so the time for a frame of data transfer depends on the number of nodes in the cluster. While the distributed algorithm for determining cluster-head nodes ensures that the expected number of clusters per round is k, it does not guarantee that there are k clusters at each round. In addition, the set-up protocol does not guarantee that nodes are evenly distributed among the cluster-head nodes. Therefore, the number of nodes per cluster is highly variable in LEACH, and the amount of data each node can send to the cluster-head varies depending on the number of nodes in the cluster.

To reduce energy dissipation, each non-cluster-head node uses power control to set the amount of transmit power based on the received strength of the cluster-head advertisement. Furthermore,

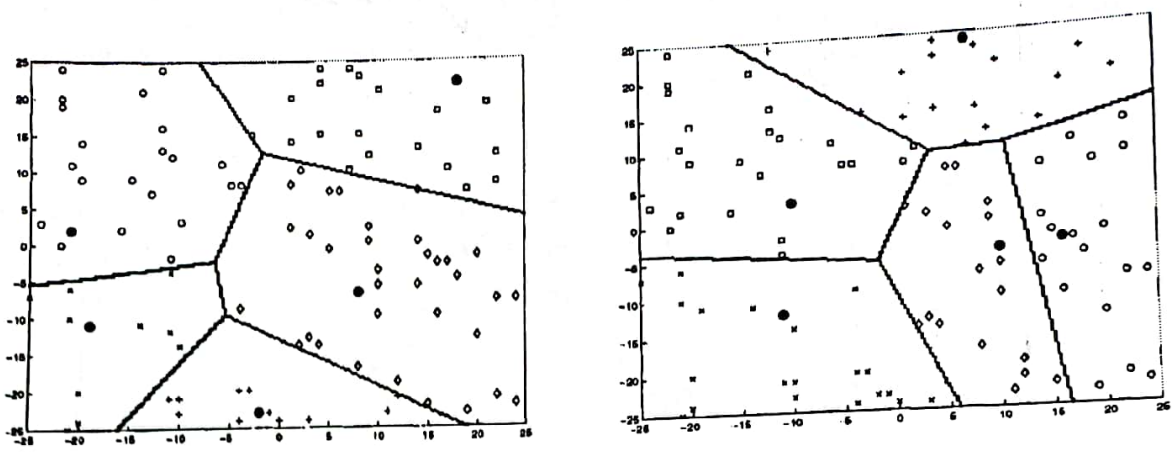


Figure 3-4: Dynamic cluster formation during two different rounds of LEACH. All nodes marked with a given symbol belong to the same cluster, and the cluster-head nodes are marked with •.

the radio of each non-cluster-head node is turned off until its allocated transmission time. Since all the nodes have data to send to the cluster-head and the total bandwidth is fixed, using a TDMA schedule is efficient use of bandwidth and represents a low latency approach, in addition to being energy-efficient.

The cluster-head must keep its receiver on to receive all the data from the nodes in the cluster. Once the cluster-head receives all the data, it can operate on the data (e.g., performing data aggregation, discussed in Section 3.3), and then the resultant data are sent from the cluster-head to the base station. Since the base station may be far away and the data messages are large, this is a high-energy transmission. Figure 3-5 shows a flow-graph of the steady-state operation.

Figure 3-6 shows the time-line for a single round of LEACH, from the time clusters are formed during the set-up phase, through the steady-state operation when data are transfered from the nodes to the cluster-heads and forwarded to the base station.

The preceding discussion describes communication within a cluster. The MAC and routing protocols were designed to ensure low energy dissipation in the nodes and no collisions of data messages within a cluster. However, radio is inherently a broadcast medium. As such, transmission in one cluster will affect (and often degrade) communication in a nearby cluster. For example, Figure 3-7 shows the range of communication for a radio, where node A's transmission, while intended for node B, collides with and corrupts any concurrent transmission intended for node C. To reduce inter-cluster interference, each cluster in LEACH communicates using direct-sequence spread spectrum (DS-SS) (described in Section 2.1.2). Each cluster uses a unique spreading code;

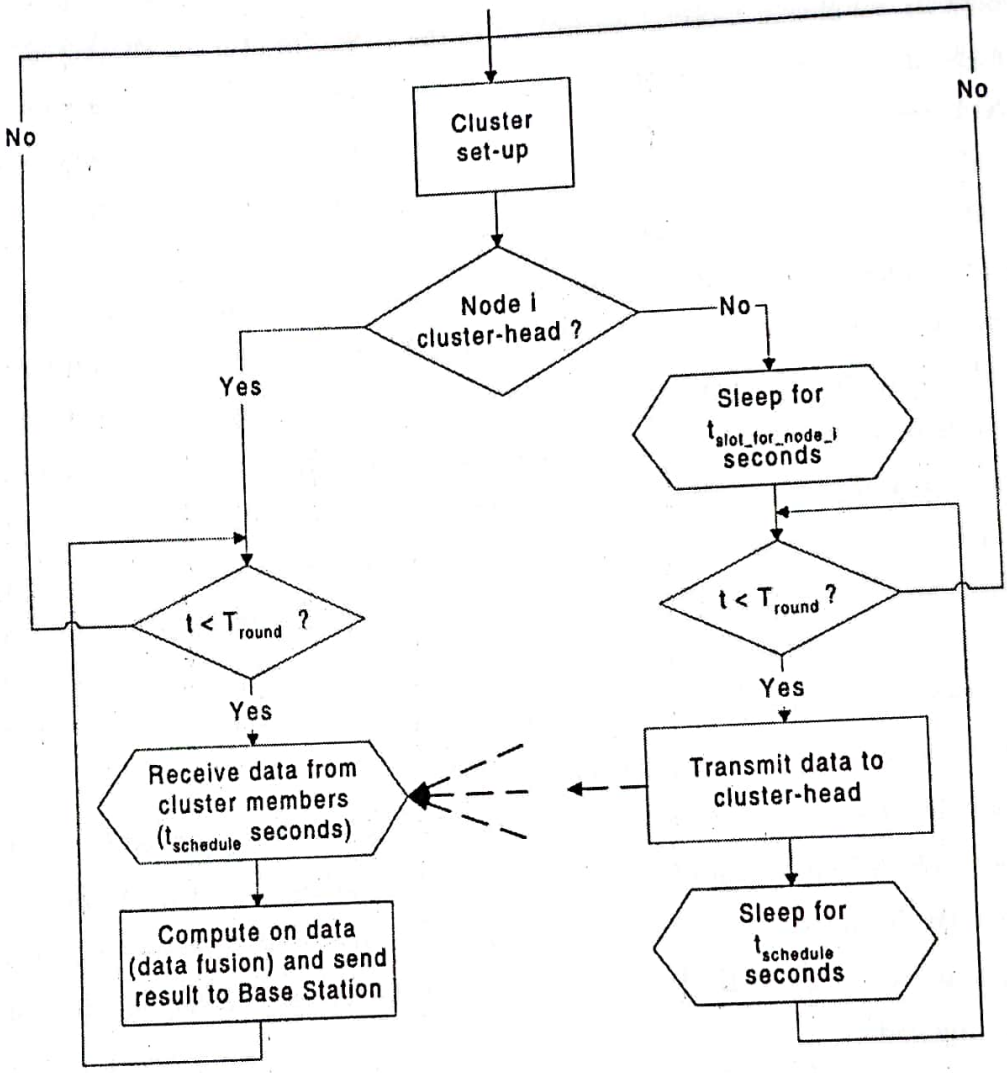


Figure 3-5: Flow-graph of the steady-state operation for LEACH.

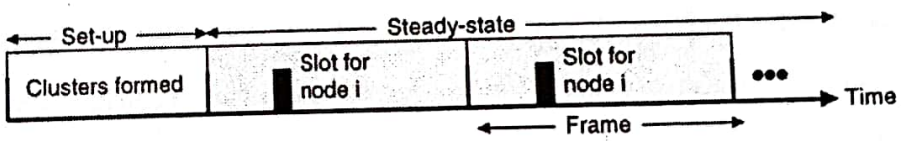


Figure 3-6: Time-line showing LEACH operation. Data transmissions are explicitly scheduled to avoid collisions and increase the amount of time each non-cluster-head node can remain in the sleep state.

all the nodes in the cluster transmit their data to the cluster-head using this spreading code and the cluster-head filters all received energy using this spreading code. This is known as *transmitter-based code assignment* [44], since all transmitters within the cluster use the same code. The first cluster-head to advertise its position is assigned the first code on a pre-defined list, the second cluster-head to advertise its position is assigned the second code, etc[2]. With enough spreading, neighboring clusters' radio signals will be filtered out as noise during de-correlation and not corrupt the transmission from nodes in the cluster. To reduce the possibility of interfering with nearby clusters and reduce its own energy dissipation, each node adjusts its transmit power. Therefore, there will be few overlapping transmissions and little spreading of the data is actually needed to ensure a low probability of collision. Note that each cluster-head only needs a single matched-filter correlator since all the signals destined for it use the same spreading code. This differs from a CDMA approach where each node would have a unique code and the base station receiver would need a bank of matched filters to obtain the data. Combining DS-SS ideas with a TDMA schedule reduces inter-cluster interference while eliminating intra-cluster interference and requiring only a single matched-filter correlator for receiving the data.

Data from the cluster-head nodes to the base station is sent using a fixed spreading code and a CSMA approach. When a cluster-head has data to send (at the end of its frame), it must sense the channel to see if anyone else is transmitting using the base station spreading code. If so, the cluster-head waits to transmit the data. Otherwise, the cluster-head sends the data using the base station spreading code.

3.3 Sensor Data Aggregation

Section 2.2 described the need to aggregate sensor data to produce a meaningful description of events that are occurring in the environment. Data aggregation can be performed on all the unprocessed data at the base station, or it can be performed locally at the cluster-heads. If the energy for communication is greater than the energy for computation, performing data aggregation locally at the cluster-head can reduce the overall system energy consumption, since much less data needs to be transmitted to the base station. This will allow large computation versus communication energy gains with little to no loss in overall network quality.

[2]If there are more clusters than spreading codes, some clusters will use the same code, possibly causing data collisions if the clusters are located close to each other.

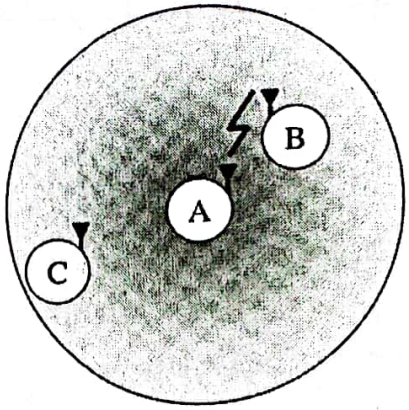


Figure 3-7: Interaction between multiple clusters. Since radio is a broadcast medium, node A's transmission, while intended for node B, collides with and corrupts any concurrent transmission intended for node C.

We can analytically compare the energy dissipation required to perform local data aggregation and send the aggregate data to the base station versus sending the unprocessed data to the base station. Suppose that the energy dissipation per bit for data aggregation is E_{DA} and the energy dissipation per bit to transmit to the base station is E_{TX}. In addition, suppose that the data aggregation method can compress the data with a ratio of L:1. This means that for every L bits that must be sent to the base station when no data aggregation is performed, only 1 bit must be sent to the base station when local data aggregation is performed. Therefore, the energy to perform local data aggregation and transmit the aggregate signal for every L bits of data is:

$$E_{Local-DA} = LE_{DA} + E_{TX} \tag{3.11}$$

and the energy to transmit all L bits of data directly to the base station is:

$$E_{No-DA} = LE_{TX} \tag{3.12}$$

Therefore, performing local data aggregation requires less energy than sending all the unprocessed data to the base station when:

$$\begin{aligned} E_{Local-DA} &< E_{No-DA} \\ LE_{DA} + E_{TX} &< LE_{TX} \\ E_{DA} &< \frac{L-1}{L} E_{TX} \end{aligned} \tag{3.13}$$

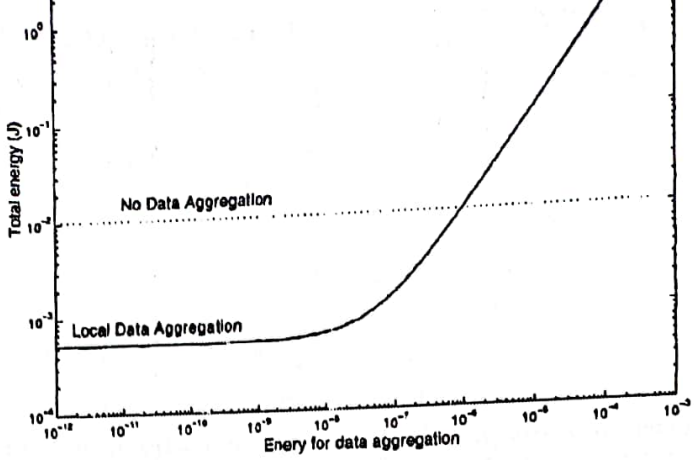


Figure 3-8: Energy dissipation to perform local data aggregation and transmit the aggregate signal to a remote base station compared with sending all the data directly to the base station as the energy cost of performing data aggregation is varied between 1 pJ/bit/signal and 1 mJ/bit/signal.

Examples

To confirm these results, we ran an experiment where $N = 20$ nodes sent data to the cluster-head and the data are either aggregated locally, requiring the cluster-head to only send a single signal to the base station ($L = 20$), or all of the unprocessed data are sent to the base station. For this simulation, the base station was 100 m away from the cluster-head node, and the cost for communicating a single bit to the base station was 1.05×10^{-6} J. Figure 3-8 shows the total energy dissipated in the system when local processing is performed and the aggregate data set is sent to the base station (labeled "Local Data Aggregation") versus the total energy dissipated in the system when the unprocessed data signals are sent to the base station (labeled "No Data Aggregation") as the energy required to perform the data aggregation functions varies between 1 pJ/bit/signal and 1 mJ/bit/signal. As expected, when the energy to perform data aggregation is less than $\frac{19}{20}1.05 \times 10^{-6} \approx 1 \times 10^{-6}$ J, the total energy dissipated in the system is less using local processing of the data. However, when the cost of aggregating the signals is higher than 1 μJ/bit/signal, it is more energy-efficient to send the data directly to the base station[3]. This computation versus communication trade-off can be made at the cluster-head node, based on models for the energy-dissipation of computation and communication.

[3]The computation model we use in our simulations, describe in Chapter 4, assumes beamforming data aggregation that consumes 5 nJ/bit/signal.

heterogenous

* ما بقدر اعمل Compress 1:1 كان Uncorrelated ما بنستفيد من Data aggregation

لازم الداتا تكون homogenous Correlated عشان اعمل Compression.

3.4 Data Correlation

In order for the cluster-head to perform data aggregation to compress the data into a single signal, data from the different nodes in the cluster must be correlated. The question we need to answer is what is the probability that an event will fall into the "view" of all the sensors in the cluster? In other words, how often can we expect to aggregate all signals from cluster members into a single signal that describes the event seen by all the nodes? Alternatively, if we aggregate all data into a single signal, what is the probability that we will miss events?

It is important to have a data-independent model for estimating the amount of correlation that exists between the data from different sensor nodes in order to estimate the amount of compression LEACH can achieve using local data aggregation. If we assume that the source signal travels a distance ρ before it can no longer be reliably detected by the sensors (due to signal attenuation), and that the sensors are omnidirectional (e.g., acoustic, seismic sensors), the maximum distance between sensors with correlated data is 2ρ, as shown in Figure 3-9a. However, being within 2ρ of each other does not guarantee that the two sensors will detect the same signal (Figure 3-9b). If all nodes are within a cluster of diameter d (i.e., the maximum distance between two nodes is d) and $d < 2\rho$, the views of the individual sensors will overlap. This implies that there will be correlation among the data from different sensors in this case. To determine the amount of correlation, we first need to find the percentage of area overlapped by j sensors in order to calculate the probability that a source is detected by j sensors within the cluster. This function f will depend on the parameters ρ and d as well as the total number of nodes in the cluster, N, and is defined as:

$$f(j, N, \rho, d) = \frac{A(j)}{A_{total}} \tag{3.14}$$

where $A(j)$ is the area overlapped by j and only j sensors' views and $A_{total} = \sum_{j=1}^{N} A(j)$ is the total area seen by at least one sensor in the cluster.

Geometrically, it is only possible to bound the area overlapped by all N sensors within a cluster of diameter d. Figure 3-10 shows the *minimum* amount of overlap, which occurs when all sensors are on the circumference of the cluster boundary and $N \to \infty$. From this figure, we see that the total area of overlap of all N nodes in the cluster, $A(N)$, in their views of source signals is:

$$A(N) = \pi(\rho - \frac{d}{2})^2 \tag{3.15}$$

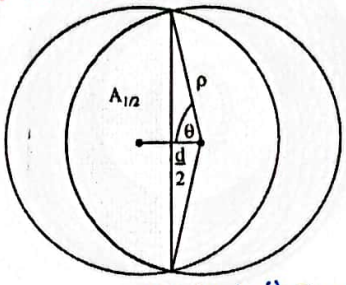


Figure 3-11: If the cluster diameter is $d = x\rho$ and each nodes' view of the world has radius ρ, then the fraction of area seen by all N nodes in the cluster is maximized when $N = 2$. In this case,

$$f(j = N, N = 2, x) = \frac{2\cos^{-1}\frac{x}{2} - x\sqrt{1-\frac{x^2}{4}}}{2\pi - 2\cos^{-1}\frac{x}{2} + x\sqrt{1-\frac{x^2}{4}}}.$$

greater than 1 (i.e., $d > \rho$), there is very little area seen by every node. In addition, $f(j = N, N, x)$ depends on the value of N; if N is small, $f(j = N, N, x)$ will be closer to the upper bound, and if N is large, it will be closer to the lower bound.

3.5 LEACH-C: Base Station Cluster Formation

The previous sections described LEACH in detail. While there are advantageous to using LEACH's distributed cluster formation algorithm, where each node makes autonomous decisions that result in all the nodes being placed into clusters, this protocol offers no guarantee about the placement and/or number of cluster-head nodes. Since the clusters are adaptive, obtaining a poor clustering set-up during a given round will not greatly affect overall performance of LEACH. However, using a central control algorithm to form the clusters may produce better clusters by dispersing the cluster-head nodes throughout the network. This is the basis for LEACH-C (LEACH-Centralized), a protocol that uses a centralized clustering algorithm and the same steady-state protocol as LEACH (e.g., nodes send their data to the cluster-head, and the cluster-head aggregates the data and sends the aggregate signal to the base station).

During the set-up phase of LEACH-C, each node sends information about its current location and energy level to the base station. The base station runs an optimization algorithm to determine the clusters for that round. The clusters formed by the base station will in general be better

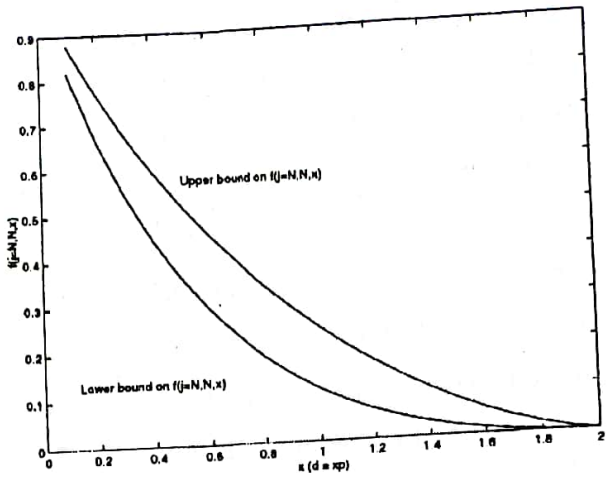


Figure 3-12: Upper and lower bounds for $f(j = N, N, x)$, where N is the number of nodes in the network, as a function of x, where $d = x\rho$ is the diameter of the cluster and ρ is the radius of each node's view of the environment.

than those formed using the distributed algorithm. However, LEACH-C requires that each node transmit information about its location to the base station at the beginning of each round. This information may be obtained by using a global positioning system (GPS) receiver that is activated at the beginning of each round to get the node's current location [61].

Determining optimal clusters from the nodes is a problem that is known to be NP-Hard [4]. Approximation algorithms, such as taboo search or simulated annealing [66], can be used to approach the optimal solution in polynomial time. Simulated annealing is an algorithm based on thermodynamics principles. If a solid material is melted and allowed to cool, the energy of the system enters several intermediate states before settling at the low-energy final state. If the system enters a state that is lower in energy than its previous state, the system remains there. However, if the system enters a state that is higher in energy than its previous state, the system remains there with a probability given by:

$$p = e^{-\frac{\Delta E}{k_{Boltz}T}} \tag{3.24}$$

where k_{Boltz} is the Boltzmann constant and T is a fixed temperature. This algorithm can be applied to optimization problems where ΔE is replaced with the difference in cost between the new state and the old state, and $k_{Boltz}T$ is a parameter that must be picked to ensure that the algorithm converges.

In addition to determining good clusters, the base station needs to ensure that the energy load

is evenly distributed among all the nodes. To do this, the base station computes the average node energy, and whichever nodes have energy below this average cannot be cluster-heads for the current round. Using the remaining nodes as possible cluster-heads, the base station runs a simulated annealing algorithm to determine the best k nodes to be cluster-heads for the next round and the associated clusters. This algorithm minimizes the amount of energy the non-cluster-head nodes will have to use to transmit their data to the cluster-head, by minimizing the total sum of squared distances between all the non-cluster-head nodes and the closest cluster-head[4]. At each iteration, the next state, which consists of a set of nodes in $C\prime$, is determined from the current state, the set of nodes in C, by randomly (and independently) perturbing the x and y coordinates of the nodes c in C to get new coordinates $x\prime$ and $y\prime$. The nodes that have location closest to $(x\prime,y\prime)$ become the new set of cluster-head nodes $c\prime$ that make up set $C\prime$. Given the current state at iteration k, represented by the set of cluster-head nodes C with cost $f(C)$, the new state, represented by the set of cluster-head nodes $C\prime$ with cost $f(C\prime)$, will become the current state with probability:

$$p_k = \begin{cases} e^{-(f(C\prime)-f(C))/\alpha_k} & : \quad f(C\prime) \geq f(C) \\ 1 & : \quad f(C\prime) < f(C) \end{cases}$$

where α_k is the control parameter (equivalent to the temperature parameter in the thermodynamic model) and $f(\cdot)$ represents the cost function defined by

$$f(C) = \sum_{i=1}^{N} \min_{c \in C} d^2(i,c) \tag{3.25}$$

where $d(i,c)$ is the distance between node i and node c. The parameter α_k must be chosen to be increasing with increasing k to ensure that the algorithm converges. However, if α_k increases too quickly, the system will get stuck in local minima. On the other hand, if α_k increases too slowly, the system will take a very long time to converge. Using simulations, we found that the following value for α_k works well for determining good clusters:

$$\alpha_k = 1000 e^{\frac{k}{20}} \tag{3.26}$$

[4]As noted previously, communication energy may not scale exactly with distance, e.g., if a building or tree is impeding a good communication channel. However, gathering information about the communication channel between all nodes is impractical. Using distance is therefore an approximation of the amount of energy that will be required for communication.

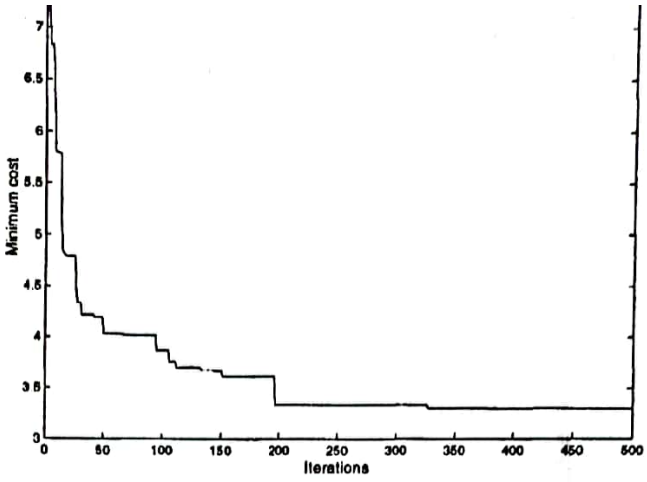


Figure 3-13: Cost function as the simulated annealing algorithm progresses. The algorithm typically converges in 200-500 iterations for a 100 node network.

Using this value of α_k, the algorithm typically converges in 200-500 iterations for a 100 node network. Figure 3-13 shows the overall decrease in the cost function as the algorithm progresses. Since these computations are being performed at the base station, energy dissipation is not a concern.

Once the optimal cluster-heads and associated clusters are found, the base station transmits this information back to all the nodes in the network. This is done by broadcasting a message that contains the cluster-head ID for each node. If a node's cluster-head ID matches its own ID, that node takes on the cluster-head role; otherwise, the node determines its TDMA slot for data transmission and goes to sleep until it is time to transmit data to its cluster-head. The steady-state phase of LEACH-C is identical to that of LEACH.

3.6 LEACH-F: Fixed Cluster, Rotating Cluster-Head

Adapting the clusters depending on which nodes are cluster-heads for a particular round is advantageous because it ensures that nodes communicate with the cluster-head node that requires the lowest amount of transmit power. In addition to reducing energy dissipation, this ensures minimum inter-cluster interference, as the power of an interfering message will be less than (or, at most, equal to) the power of the message the cluster-head is receiving (see Figure 3-14). If, on

متاكله :
* انه لازم كلهم نفس الـ energy اولما .
* المشكلة اذا Sensor ← مثلا رح تكون في مشكلة بالشبكة لانه في Nodes بجاد موكلهم نفس البجد .
او اذا دخلت Sensor جديد (Mobile)

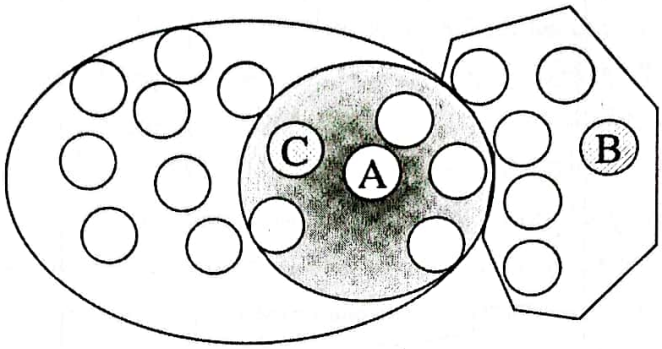


Figure 3-14: If the clusters are adaptive and change depending on the location of the cluster-head nodes, there is only minimal inter-cluster interference. In this figure, node A chooses to join cluster-C because it requires less transmit power to communicate with node C than node B, the other choice for cluster-head. In addition to minimizing the non-cluster-head nodes' energy dissipation, adaptive clustering reduces inter-cluster interference.

the other hand, the clusters were fixed and only the cluster-head nodes were rotated, a node may have to use a large amount of power to communicate with its cluster-head when there is another cluster's cluster-head close by. For example, in Figure 3-15, node A needs to use a large amount of transmit power to communicate with its cluster-head, node B. Since cluster-head C is located close to node A, node A's transmission will corrupt any transmission to cluster-head C. Therefore, using fixed clusters and rotating cluster-head nodes within the cluster may require more transmit power from the nodes, increasing non-cluster-head node energy dissipation and increasing inter-cluster interference.

The advantage of fixed clusters is that once the clusters are formed, there is no set-up overhead at the beginning of each round. Depending on the cost for forming adaptive clusters, an approach where the clusters are formed once and fixed and the cluster-head position rotates among the nodes in the cluster may be more energy-efficient than LEACH. This is the basis for LEACH-F (LEACH with Fixed clusters). In LEACH-F, clusters are created using the centralized cluster formation algorithm developed for LEACH-C. The base station uses simulated annealing to determine optimal clusters and broadcasts the cluster information to the nodes. This broadcast message includes the cluster ID for each node, from which the nodes can determine the TDMA schedule and the order to rotate the cluster-head position. The first node listed in the cluster becomes cluster-head for the first round, the second node listed in the cluster becomes cluster-head for the second round, and

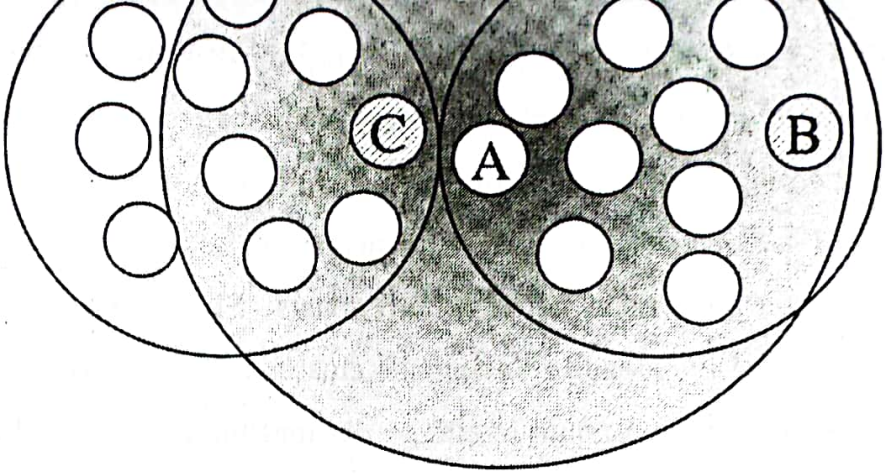


Figure 3-15: If the clusters are fixed and only the cluster-head nodes are rotated, there can be significant inter-cluster interference. In this figure, node A has to use a large amount of transmit power to communicate with its cluster-head, node B. Since cluster-head C is much closer to node A than cluster-head B, node A's transmission will cause a large amount of interference to any transmissions cluster-head C is receiving from its cluster members.

so forth. Using LEACH-F, there is no set-up required for the different rounds—nodes implicitly know when they are cluster-heads and when they are non-cluster-heads[5]. The steady-state phase of LEACH-F is identical to that of LEACH.

LEACH-F would not be practical in any sort of dynamic system. The fixed nature of this protocol does not allow new nodes to be added to the system and does not adjust its behavior based on nodes dying. Furthermore, LEACH-F does not handle node mobility. Therefore, while this is a good comparison protocol to determine the advantage of a no-overhead approach, it may not be a useful protocol architecture for real systems.

3.7 Summary

This chapter introduced our cross-layer protocol architecture, LEACH. LEACH was designed to exploit the application-specific function of sensor networks, where the end-user requires information about events occurring in the environment, rather than nodes' individual data. In addition, LEACH

[5]In a practical system, there would probably be some set-up at the beginning of each round to ensure the nodes in the cluster are all time-synchronized.

Chapter 4

Analysis and Simulation of LEACH

For even moderately-sized networks with tens of nodes, it is impossible to analytically model the interactions between all the nodes. Therefore, simulation was used to determine the benefits of different protocols. Computation and communication energy dissipation models as well as new MAC algorithms were implemented in ns to support the design and simulation of the different protocol architectures. In the experiments described in this chapter, LEACH is compared with LEACH-C (the centralized set-up algorithm), LEACH-F (the fixed cluster, rotating cluster-head algorithm), MTE routing (where data traverses multiple short hops to reach the base station), and static clustering (where clusters and cluster-head nodes are fixed) in terms of system lifetime, energy dissipation, amount of data transfer (actual data for MTE routing, aggregate data for the LEACH protocols), and latency.

4.1 Simulation Models

In order to compare different protocols, it is important to have good models for all aspects of communication. This section describes the models that were used for channel propagation, communication energy dissipation, and computation energy dissipation.

4.1.1 Channel Propagation Model

In a wireless channel, the electromagnetic wave propagation can be modeled as falling off as a power law function of the distance between the transmitter and receiver. In addition, if there is no direct, line-of-sight path between the transmitter and the receiver, the electromagnetic wave will bounce off objects in the environment and arrive at the receiver from different paths at different times.

This causes multipath fading, which again can be roughly modeled as a power law function of the distance between the transmitter and receiver. No matter which model is used (direct line-of-sight or multipath fading), the received power decreases as the distance between the transmitter and receiver increases [75].

For the experiments described in this dissertation, both the free space model and the multipath fading model were used, depending on the distance between the transmitter and receiver, as defined by the channel propagation model in ns [14, 75]. If the distance between the transmitter and receiver is less than a certain cross-over distance ($d_{crossover}$), the Friss free space model is used (d^2 attenuation), and if the distance is greater than $d_{crossover}$, the two-ray ground propagation model is used (d^4 attenuation). The cross-over point is defined as follows:

$$d_{crossover} = \frac{4\pi\sqrt{L}h_r h_t}{\lambda} \tag{4.1}$$

where

$L \geq 1$ is the system loss factor not related to propagation,

h_r is the height of the receiving antenna above ground,

h_t is the height of the transmitting antenna above ground, and

λ is the wavelength of the carrier signal.

If the distance is less than $d_{crossover}$, the transmit power is attenuated according to the Friss free space equation as follows:

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi d)^2 L} \tag{4.2}$$

where

$P_r(d)$ is the receive power given a transmitter-receiver separation of d,

P_t is the transmit power,

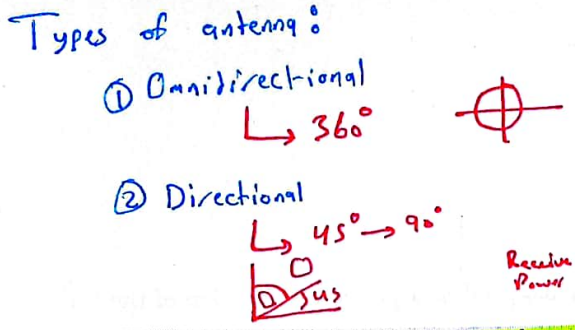
G_t is the gain of the transmitting antenna,

G_r is the gain of the receiving antenna,

λ is the wavelength of the carrier signal,

d is the distance between the transmitter and the receiver, and

$L \geq 1$ is the system loss factor not related to propagation,



This equation models the attenuation when the transmitter and receiver have direct, line-of-sight communication, which will only occur if the transmitter and receiver are close to each other (i.e., $d < d_{crossover}$). If the distance is greater than $d_{crossover}$, the transmit power is attenuated according to the two-ray ground propagation equation as follows:

$$P_r(d) = \frac{P_t G_t G_r h_t^2 h_r^2}{d^4} \tag{4.3}$$

where

$P_r(d)$ is the receive power given a transmitter-receiver separation of d,

P_t is the transmit power,

G_t is the gain of the transmitting antenna,

G_r is the gain of the receiving antenna,

h_r is the height of the receiving antenna above ground,

h_t is the height of the transmitting antenna above ground, and

d is the distance between the transmitter and the receiver.

In this case, the received signal comes from both the direct path and a ground-reflection path [75]. Due to destructive interference when there is more than one path through which the signal arrives, the signal is attenuated as d^4.

In the experiments described in this dissertation, an omnidirectional antenna was used with the following parameters: $G_t = G_r = 1$, $h_t = h_r = 1.5$ m, no system loss ($L = 1$), 914 MHz radios, and $\lambda = \frac{3\times10^8}{914\times10^6} = 0.328$ m. Using these values, $d_{crossover} = 86.2$ m and Equations 4.2 and 4.3 simplify to:

$$P_r = \begin{cases} 6.82 \times 10^{-4} \frac{P_t}{d^2} & : \; d < 86.2 \text{ m} \\ 2.25 \frac{P_t}{d^4} & : \; d \geq 86.2 \text{ m} \end{cases} \tag{4.4}$$

4.1.2 Radio Energy Model

There has been a significant amount of research in the area of low-energy radios. Different assumptions about the radio characteristics, including energy dissipation in the transmit and receive modes, will change the advantages of different protocols. In this work, we assume a simple model where the transmitter dissipates energy to run the radio electronics and the power amplifier and the receiver dissipates energy to run the radio electronics [88]. As discussed in the previous section,

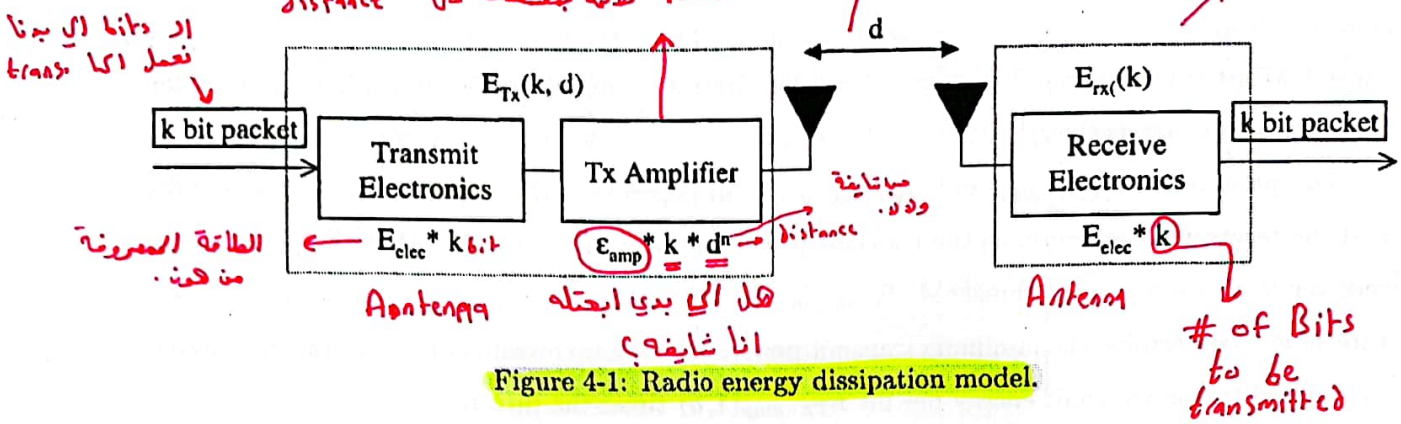


Figure 4-1: Radio energy dissipation model.

the power attenuation is dependent on the distance between the transmitter and receiver. For relatively short distances, the propagation loss can be modeled as inversely proportional to d^2, whereas for longer distances, the propagation loss can be modeled as inversely proportional to d^4. Power control can be used to invert this loss by setting the power amplifier to ensure a certain power at the receiver. Thus, to transmit an l-bit message a distance d, the radio expends:

$$E_{Tx}(l,d) = E_{Tx-elec}(l) + E_{Tx-amp}(l,d) \tag{4.5}$$

$$E_{Tx}(l,d) = \begin{cases} lE_{elec} + l\epsilon_{friss-amp}d^2 & : d < d_{crossover} \\ lE_{elec} + l\epsilon_{two-ray-amp}d^4 & : d \geq d_{crossover} \end{cases} \tag{4.6}$$

and to receive this message, the radio expends:

$$E_{Rx}(l) = E_{Rx-elec}(l)$$

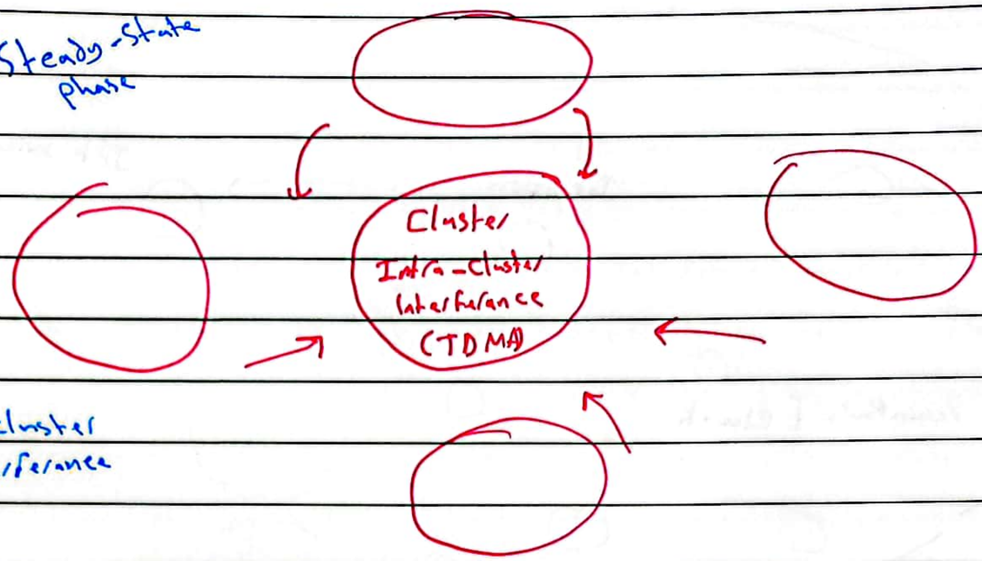
$$E_{Rx}(l) = lE_{elec} \tag{4.7}$$

as shown in Figure 4-1. The electronics energy, E_{elec} depends on factors such as the digital coding, modulation, and filtering of the signal before it is sent to the transmit amplifier. In addition, when using DS-SS, the electronics energy accounts for the spreading of the data when transmitting and the correlation of the data with the spreading code when receiving. Researchers have designed transceiver baseband chips that support multi-user spread-spectrum communication and operate at 165 mW in transmit mode and 46.5 mW in receive mode [83]. For the experiments described in this dissertation, we set the energy dissipated per bit in the transceiver electronics to be

$$E_{elec} = 50 \text{ nJ/bit} \tag{4.8}$$

14/11

Steady-State phase



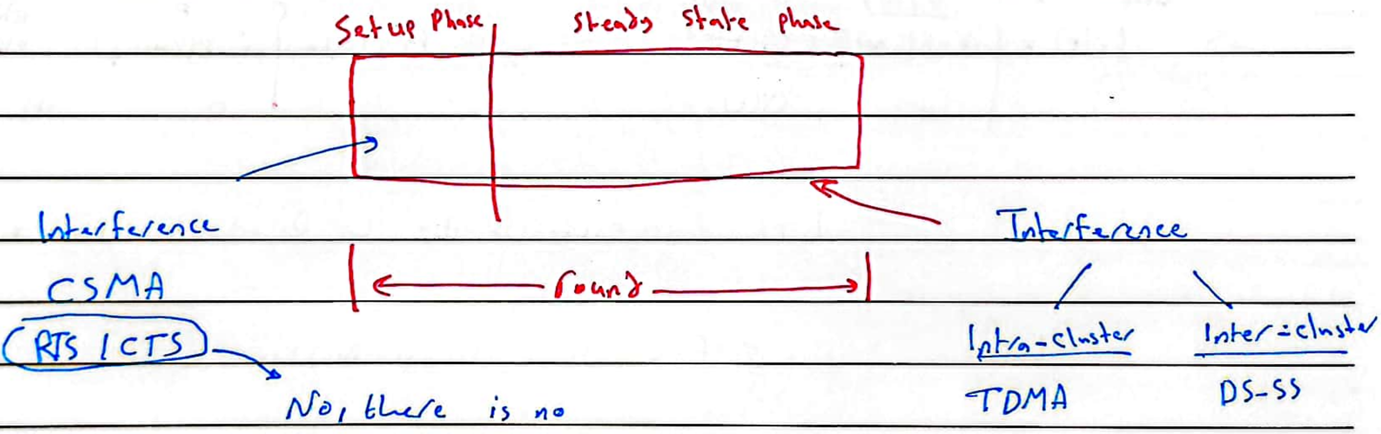
Inter-cluster Interferance

- CDMA
- Direct Sequence spread spectrum (DS-SS)

* حل الـ Intra-cluster Interferance عن طريق الـ TDMA .

Unique spreading code for all cluster's members

* بصير اعرف الـ ID الـ بوصلني مرا كل عشان الكود بختلف



No, there is no master a CH selected at this moment

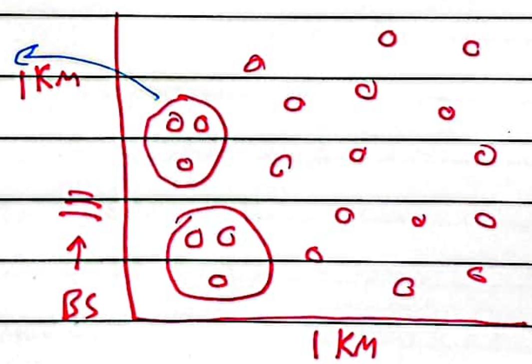
Slide 73

قراب على الـ base فمش فرقت صار عالفاضي

* بمنا متحيل عدد الـ Cluster الافضل

وهديه ورينا مكان عن CH الـ cluster

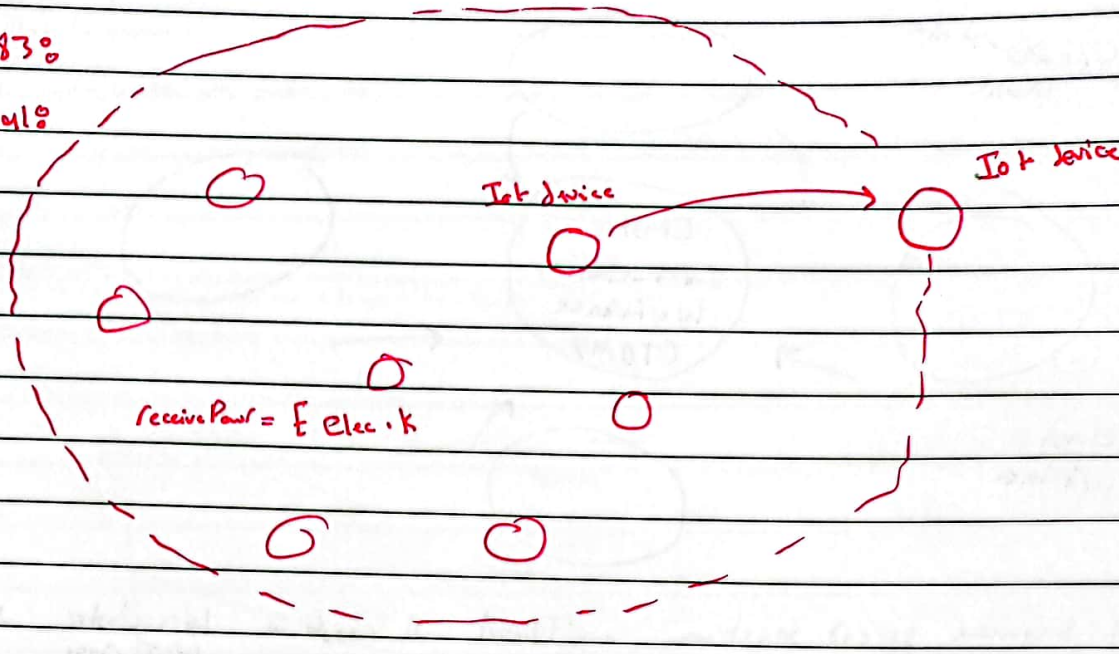
هيك طلع عنا LEACH-C



16/11

Slide 83

Omnidirectional



MT-CHR

* كيف يعمل Re-cluster ← every base Round

* يعمل Re-cluster لما يقل الـ energy عن E_{Th}

* كل ما اعمل Read Req: بدي ازيد الـ E_{Th} اكتر شوي

* CountRND قديش عدد الـ Round اللي بدي اضم فيها.

* كل ما اوصل انا لهاد الـ E_{Th} معناتو الوضع خطير وبدي reclaster فبزيد الـ E_{Th}.

$$E_{Th} = CountRND \times P_{TX} \times n$$

$$CountRND = \frac{P_{TL}}{Pwec} \times 100$$

of nodes that are eligible to be cluster heads in round r, $C_i(t)=1$ means that the node i is eligible while $C_i(t)=0$ means that node i is not eligible in this round as it has served as a cluster head recently. It can be extracted from this probability that after $\frac{N}{k}$ rounds, all nodes should serve as cluster heads. Therefore, the opportunity for nodes, that did not serve as cluster heads yet, to become cluster heads increases as the round approaches $\frac{N}{k}$. If we let P represents the percentage of the expected number of cluster heads in the network, which is equal to $(\frac{k}{N})$, then dividing the numerator and denominator of Eq. (1) by N yields [22]:

$$P_{Th}(r) = \begin{cases} \frac{P}{1-P(r \bmod \frac{1}{P})}, & \text{if a node is eligible} \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

where $P_i(t)$ has termed as threshold probability $P_{Th}(r)$.

One of the major drawbacks of LEACH protocol is the control overhead encountered in every round. T-LEACH protocol addresses this and subsequently proposes a model in which the control overhead is significantly reduced. In detail, the cluster heads do not change every round but rather every batch of rounds. The cluster heads will remain serving as long as their energy does not dip below a certain proposed threshold energy that is basically formulated in a way does not exceed the transmit power needed for a node to send its data to the cluster head. As introduced in T-LEACH protocol, the threshold energy is calculated per cluster upon the following:

$$E_{Th} = Count_{RND} \times P_{TX} \times n, \quad (3)$$

where,

$$Count_{RND} = \frac{P_{HR}}{P_{WEC}} \times 100, \quad (4)$$

and P_{TX} represents the power of transmitting 1 byte data, n is the data message length (in bytes) for any member node, P_{HR} is the head replacement cost (i.e., the energy consumed to elect a new cluster head which includes announcement message, join queries, and TDMA schedule), and P_{WEC} represents whole energy in the cluster. If the initial energy of all nodes is similar, then P_{WEC} can be found by multiplying this value by the number of nodes per cluster. Accordingly, $Count_{RND}$ represents the times of round.

As opposed to utilizing a heuristic approach, the authors in [29] proposed a disseminated LEACH-based cluster head determination calculations, namely LEACH with Distance-based Thresholds (LEACH-DT) in which sensor nodes are self-chosen to wind up CHs with various probabilities in light of their separations to the BS, such that the vitality utilization among these nodes are adjusted. This complies with the first LEACH convention where every one of the nodes is self-chosen to end up a CH with a consistent predesigned likelihood, and hence is imperfect. Simulation results demonstrate that LEACH-DT outperforms LEACH with enhanced system life expectancy of more than 10%.

1.2. Our methodology and contributions

Our work addresses the two limitations of T-LEACH protocol. The first one, the probability of cluster-head selection remains as is and this leads to making the clusters more energy-intensive than those in LEACH protocol. This is proven while demonstrating the proposed protocol. The other one refers to the threshold energy. In other words, it is expressed to be too small for the sake of reducing the setup overhead as much as possible. In fact, this will work against the major objective of LEACH protocol which refers to evenly distribute the energy among nodes for the reason of delaying the first node death. Besides, there will be some data loss as cluster heads will not be ultimately capable of transferring the data to the base station. In this article, a Modified Threshold-based Cluster Head Replacement (MT-CHR) protocol is proposed in which the correct probability of serving as a cluster head is introduced. Moreover, new threshold energy is proposed which basically takes just mentioned drawbacks into consideration. In fact, preliminary results of this protocol have been presented in [30].

The rest of paper is organized as follows: Section 2 illustrates the proposed algorithm. Section 3 presents the simulation environment, results, and necessary discussions. Section 4 concludes our work.

2. The proposed protocol

21. Network assumptions

The assumptions used in this article are as follows:

- Sensors are homogeneous and have the same initial energy.
- Sink node (base station) is fixed and located outside the area of concern and all sensor nodes can communicate directly with the sink node.
- Sensors are distributed uniformly in a square area.
- Sensors are not mobile (stationary) and each sensor node knows its location.
- Sensors have symmetric radio energy consumption model (i.e. sending data from node A to node B dissipates the same amount of energy when sending the same amount of data back in the reverse direction).

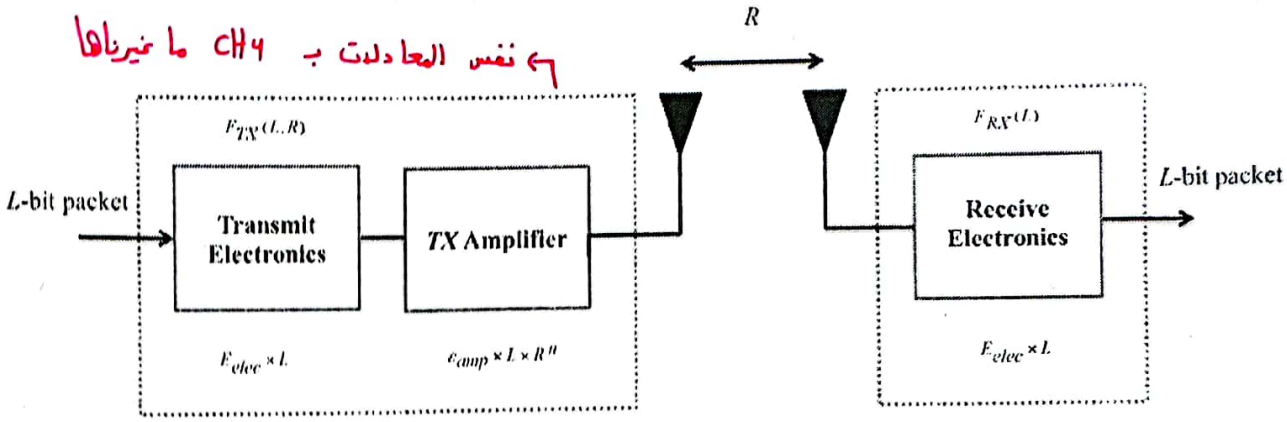


Fig. 1. Sensor radio energy and propagation models [20–22,28,29].

- All member nodes sense the monitored area constantly and at a fixed rate, thus, nodes always have data to send to the sink node.
- The computation energy is ignored as it is relatively smaller than the communication overhead [20–22,28,29].

2.2. Radio energy and channel propagation models

Nowadays, there are a tremendous number of researches carried out on energy-limited devices. Different radio characteristics such as transmitter-receiver mode and energy dissipation affect the behavior of each protocol. Consequently, it is important to focus on radio energy and channel propagation models' assumptions while designing wireless sensor protocols. Throughout the simulations of this work, the models adopted are shown in Fig. 1 and the same as those used in [20–22,28,29]. In the radio energy model, power is mainly dissipated on running transmitting and receiving electronics and further amplification, which is shown at the transmitter side. To run the electronic circuitry at both transmitter and receiver, E_{elec} is required and this is per bit. To process a L-bit message, both sender and receiver circuits consume $E_{TX-elec}(L) = E_{RX-elec}(L) = L \times E_{elec}$. Furthermore, the transmitter consumes $E_{TX-amp}(L, R)$ to amplify the signal that is to be transmitted over a distance R. In the channel propagation model, both free space and multipath fading (two-ray) electromagnetic wave propagation models are considered. In specific, if the distance, between the transmitter and receiver, is less than a threshold, denoted by R_0, then the free space model is used. Otherwise, the multipath fading model is used. In these propagation models, power control can set $E_{TX-amp}(L, R)$ amplification value to either $L\varepsilon_{fs}R^2$, for free space model, or $L\varepsilon_{mp}R^4$, for two-ray model. Therefore, the total energy, consumed at the transmitter node ($E_{TX}(L,R)$) can be expressed by:

$$E_{TX}(L,R) = \begin{cases} L\left(E_{elec} + \varepsilon_{fs}R^2\right), & R < R_0 \\ L\left(E_{elec} + \varepsilon_{mp}R^4\right), & R \geq R_0 \end{cases} \quad (5)$$

where R_0 is represented by the following:

$$R_0 = \frac{4 \times \pi \times \sqrt{l} \times h_t \times h_r}{\lambda} \quad (6)$$

where l is the loss factor, h_t defines the distance between the transmitter and ground, h_r is the distance between the receiver and ground, and λ symbolizes the carrier wavelength.

2.3. Preliminaries

To fully understand the major contribution of this work, we want to demonstrate the operations of LEACH and T-LEACH protocols along with highlighting their shortcomings. Therefore, assume that 60 sensor nodes have been deployed in a network, in which the expected number of clusters is 6, resulting that the number of nodes per cluster is $(\frac{N}{k}) = 10$ nodes. This means that after 10 rounds, all nodes should have served as cluster heads. Additionally, the same policy will be considered for the next 10 rounds. In other words, after the multiple of 10 (i.e., 20, 30, 40, ..., etc.), all nodes should have served as cluster heads. According to the deep understanding of the authors to LEACH protocol and to prove the inadequacy of T-LEACH protocol, which will be detailed shortly, Table 1 is provided. In fact, this discussion has not been mentioned in the literature at all. However, Table 1 shows the following values in each round: (1) the expected number of cluster heads that has been selected in a round within the same cluster which is symbolized by *CHS* and is equal to (rmod(1/*P*), (2) The number of participating nodes, in cluster head election, abbreviated by *PN*, (3) participating nodes percentage which is represented by *PNP* and mainly equal to (1-*P**(rmod(1/*p*)) or $(\frac{PN_{arroundr}}{N/k})$, (4) the threshold probability (P_{Th}) which is obviously formulated by $(\frac{P}{PNP})$. It is noticed that the number of nodes that can participate in clustering operations (*PN*) and its corresponding percentage (*PNP*) decrease every round whereas the threshold probability increases till round $(\frac{N}{k})$ which is completely true.

Table 1
$P_{Th}(r)$ for LEACH protocol.

Round number	CHS	PN	PNP	P_{Th}
0	0	10	1	0.1
1	1	9	0.9	0.111
2	2	8	0.8	0.125
3	3	7	0.7	0.143
4	4	6	0.6	0.167
5	5	5	0.5	0.2
6	6	4	0.4	0.25
7	7	3	0.3	0.333
8	8	2	0.2	0.5
9	9	1	0.1	1
10	0	10	1	0.1

Table 2
$P_{Th}(r)$ for T-LEACH protocol.

Round number	VRC	ACH	CHS	PN	PNP	P_{Th}
0	0	0	0	10	1	0.1
1	1	1	1	9	0.9	0.111
2	1	1	2	9	0.8	0.125
3	1	1	3	9	0.7	0.143
4	1	1	4	9	0.6	0.167
5	1	1	5	9	0.5	0.2
6	1	1	6	9	0.4	0.25
7	1	1	7	9	0.3	0.333
8	1	1	8	9	0.2	0.5
9	1	1	9	9	0.1	1
10	1	1	0	9	1	0.1
11	1	1	1	9	0.9	0.111
12	1	1	2	9	0.8	0.125
13	1	1	3	9	0.7	0.143
14	1	1	4	9	0.6	0.167
15	1	1	5	9	0.5	0.2
16	2	2	6	8	0.4	0.25
17	2	2	7	8	0.3	0.333
18	2	2	8	8	0.2	0.5
19	2	2	9	8	0.1	1

At this round (i.e., 10), all nodes should have become cluster heads. Once the *PN* gets to 1, then the threshold probability is 1 which compels the node to elect itself as a cluster head.

In T-LEACH protocol, the same procedure will be carried out by each node except that nodes in T-LEACH protocol do not collaborate in re-clustering operation every round. Hence, in a network, the number of nodes that serve as cluster heads will be fixed till the next batch of rounds. In T-LEACH protocol, re-clustering operation occurs only when the energy of a cluster head dips below a threshold energy that will be discussed in a little while. In fact, *PN* will be fixed until the next re-clustering operation. In order to highlight the imperfections of T-LEACH protocol, the authors propose a new parameter called the actual number of cluster heads selected, which is abbreviated by *ACH*. Consequently, Table 2 shows the values of this parameter along with aforementioned parameters, namely, *CHS*, *PN*, *PNP*, and P_{Th}. It deserves mentioning that T-LEACH protocol does not take the actual selected number of cluster heads into consideration in calculating *CHS*. In fact, *CHS* will stay reliant on *(r mod (1/p))* formula. This is evident in Table 2 in which *PN* is fixed and P_{Th} increases in every round. It can be seen that the threshold probability stay subject to *PNP* which in turn does not represent the genuine percentage of nodes participating in the re-clustering operation. Assuming that a cluster head energy falls under the threshold in round 15 (i.e. at round 16, new setup occurs), then *PNP* is equal to 0.5 but the actual percentage of nodes participating in the re-clustering operation is 0.9 and the threshold probability of each node to be a cluster head is 0.2 instead of 0.111 (the correct value).

Interestingly, as the number of participating nodes increases more and more, the threshold probability should acts inversely (i.e., should decrease more and more). Unfortunately, this is not achieved as seen in Table 2. In other words, when the values of *PN* are constants, then the P_{Th} values should be fixed and truly this is not the case in this table. Furthermore, the expected number of elected cluster heads per round and per cluster should be equal to 1. In different words, *PN* * P_{Th} should be equal to 1 to comply with the assumptions of LEACH protocol. If we get back to all entries in Table 1, the results always equal 1. Unluckily, this is not achieved in Table 2. Considering round 15, for example, the expected number of elected cluster heads per cluster and round (i.e., *PN* * P_{Th}) is equal to 2 ($\lceil 9*0.2 \rceil$) nodes instead of 1 node (9*0.111). Accordingly, the number of cluster heads selected increases which results in increasing the number of long-distance and direct communications with the base station and eventually increasing the energy consumption associated with re-clustering overhead.

تطوير الدكتور ←

Table 3
P_{Th}(VRC) for MT-CHR protocol.

Round number	ACH	CHS	PN	PNP	P_{Th}
0	0	0	10	1	0.1
1	1	1	9	0.9	0.111
2	1	1	9	0.9	0.111
3	1	1	9	0.9	0.111
4	1	1	9	0.9	0.111
5	1	1	9	0.9	0.111
6	1	1	9	0.9	0.111
7	1	1	9	0.9	0.111
8	1	1	9	0.9	0.111
9	1	1	9	0.9	0.111
10	1	1	9	0.9	0.111
11	1	1	9	0.9	0.111
12	1	1	9	0.9	0.111
13	1	1	9	0.9	0.111
14	1	1	9	0.9	0.111
15	1	1	9	0.9	0.111
16	2	2	8	0.8	0.125
17	2	2	8	0.8	0.125
18	2	2	8	0.8	0.125
19	2	2	8	0.8	0.125

فيما limitation فتم تطويرها

LEACH-F هاد LEACH-C لحل مشاكل الـ

.Batch كل الـ Setup بصير round مش كل

Setup Phase ويخفف من الـ distributed عشان بده يفل

re-clustering every Batch

بخدم كذا round ولما اشوف الـ energy بتقل بصير ساعتها re-cluster

وبعت للكل يعمل Synchronise عشان الـ re-cluster.

2.4. The proposed protocol: MT-CHR

To overcome aforementioned T-LEACH limitations, the modified T-LEACH, namely, MT-CHR protocol is proposed. In this protocol, a new parameter has been incorporated into the threshold probability, called Virtual Rounds Count (VRC), showing the true number of re-clustering operations carried out by each node. In other words, this parameter increases every time a node participates in cluster reformation. Therefore, the threshold probability becomes as follows:

$$P_{Th}(VRC) = \begin{cases} \frac{P}{1-P(VRC \bmod \frac{1}{P})}, & \text{if a node is eligible} \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

أول اجراء ← اخرنا مرة الـ node بالنهاية

Note that $(VRC \bmod \frac{1}{P})$ or $(VRC \bmod \frac{N}{k})$, appears at the denominator of (7), represents the cluster heads that have been selected till round (r) within the same cluster. The authors believe this is more reasonable than the old definition, as introduced in T-LEACH protocol, which is formulated by $(r \bmod \frac{1}{P})$. Furthermore, the *PNP*, in MT-CHR protocol, is expressed by $1 - P(VRC \bmod \frac{1}{P})$, which is different than the one expressed in T-LEACH protocol. Table 3 shows the MT-CHR results for *CHS*, *PN*, and P_{Th}. It can be noticed that the values of *PN*, *PNP*, P_{Th} will remain constant until the next cluster reformation operation, in which *VRC* changes. Furthermore, in all entries of this table, the expected number of elected cluster heads per cluster and round (i.e., $PN * P_{Th}$) will remain 1 and this agrees fairly with the LEACH assumptions.

When analyzing the T-LEACH threshold energy, presented in Eq. (3), we can point out the following: Firstly, as T-LEACH is a distributed protocol, the control overhead cost (P_{HR}) is relatively very small compared to the overall cluster energy (P_{WEC}). Thus, $Count_{RND}$ will approach very small value which makes it countless (not affecting). Secondly, as the steady-state cost per round is excluded from Eq. (3), it is not fair to consider $Count_{RND}$ as the times of round. It is further of interest to mention that T-LEACH threshold energy excludes the expense of data messages aggregation and sending to the base station which result in having excessively low threshold energy. This concludes that the objective of T-LEACH protocol is to make the setup overhead very low (i.e., reclustering happens rarely). Engrossingly, this will work against the major objective of LEACH protocol which is to make the first node to die very late. T-LEACH protocol may contribute, over LEACH protocol, in having a longer death of the last node. Lastly, the expression of threshold energy is byte-based which induces to lessening it more.

In MT-CHR protocol, additional amendment is proposed to the threshold energy for the sake of slowing down the cluster head energy depletion. In particular, the MT-CHR protocol has raised the threshold energy through adding, to the former threshold energy, five times the approximated data receiving and sending energy (ADRSE). Many simulations have been conducted to determine the best factor to be incorporated with the ADRSE considering a range of (1–10) where the value 5 has effectively maximized most of the performance metrics that will be discussed shortly keeping in mind that if this factor is selected near 1, the network performance will come closer to T-LEACH, while, if it is selected close to 10, the network will operate approximately similar to LEACH protocol. Intriguingly, in MT-CHR protocol, nodes that have turned into a cluster head will not disseminate their energy very soon. Actually, nodes will have enough energy to get the opportunity to participate as a cluster head or a member node in the network several times which brings on expanding the network lifetime and presents more load-balancing through the network. Thus, the threshold energy is found as follows:

$$E_{Th} = Count_{RND} \times E_{TX} \times q + 5 \times ADRSE, \quad (8)$$

تاني اجراء، (رفعنا الـ threshold) ← صار الـ node ما تموت بسرعة بالبداية

صارت بدل bit بالـ byte هاد الفرق الأول

5 أضعاف الـ E (energy) اني ابعت واستقبل للـ base-station عشان ارفع E_{th}. هاد الفرق الثاني

5 عشان لا يكون قريب من الـ T-LEACH و ١٠ قريب من الـ LEACH فطلع المناسب 5.

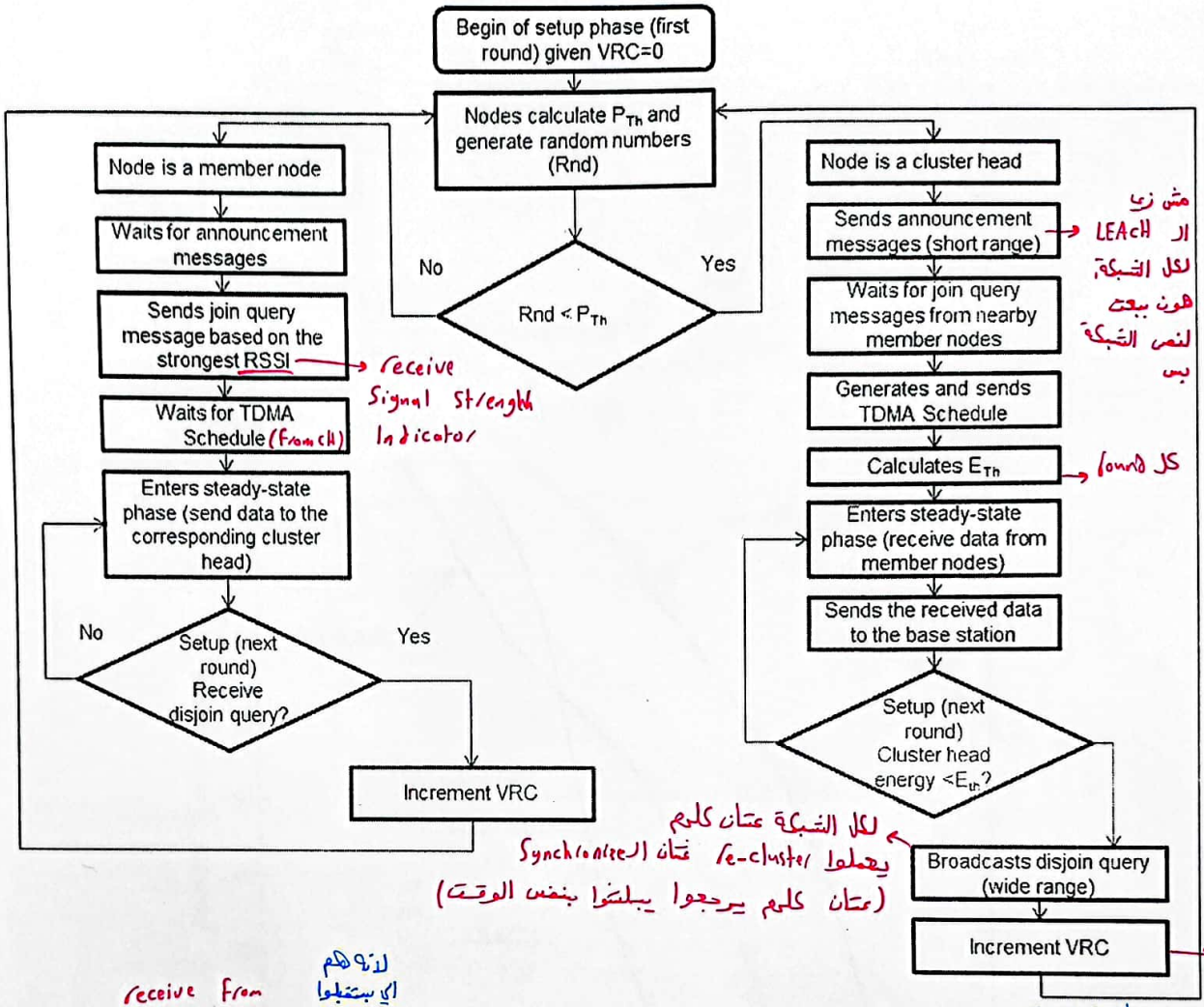


Fig. 2. The operation of MT-CHR protocol.

where,

$$ADRSE = q \times E_{RX-elec} \times \left(\frac{N}{k} - 1\right) + q \times \frac{N}{k} \times \left(E_{TX-elec} + \varepsilon_{amp} d^{m}\right) \quad (9)$$

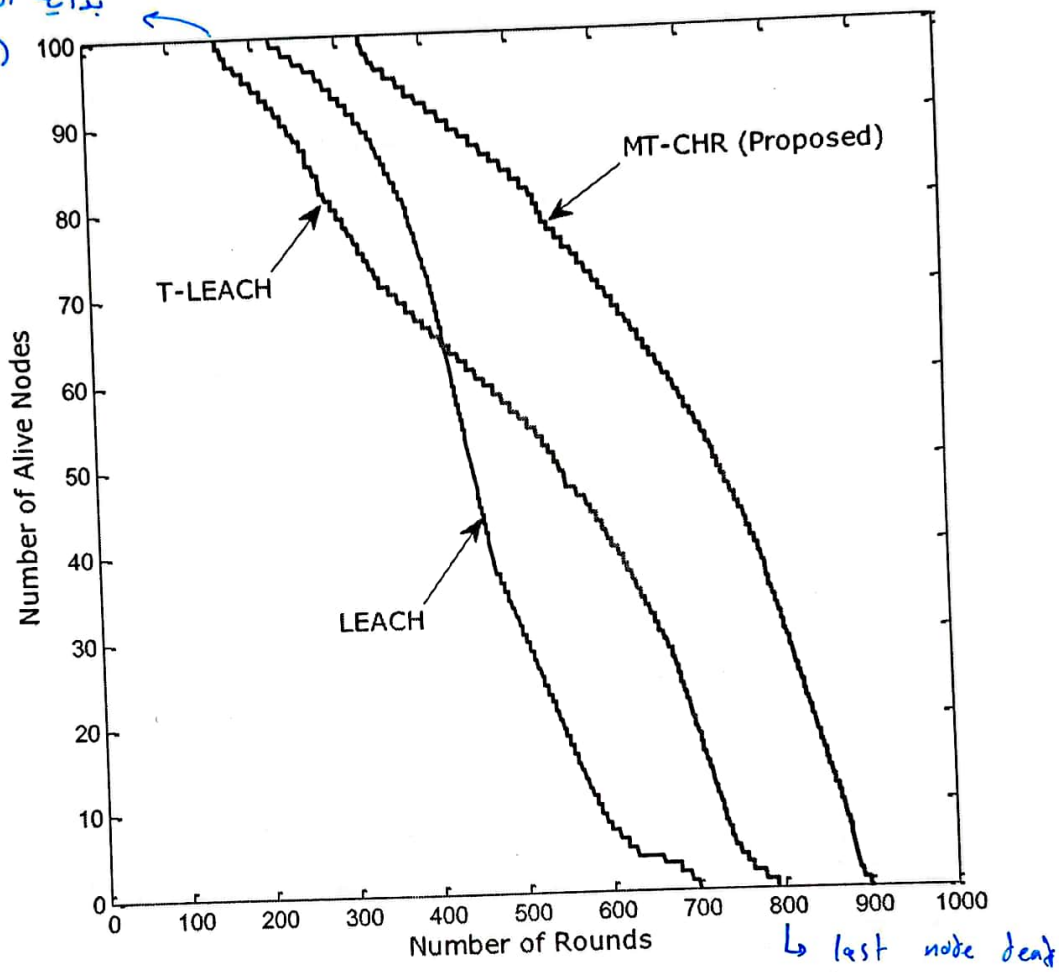
where the first term of Eq. (8) is a bit-based expression instead of byte-based as expressed in T-LEACH protocol. However, the first term of Eq. (9) refers to the receive power per cluster while the other one represents the transmit power to the base station.

Fig. 2 illustrates the complete operation of the MT-CHR protocol. Particularly, in the setup phase, nodes find P_{Th}, based on the expression introduced in Eq. (7). Accordingly, every node generates a random number between 0 and 1. If the random number generated is less than the node P_{Th}, then that node selects itself as a cluster head and broadcasts an announcement message to all nearby nodes. To avoid the collision with other possible cluster heads, this message is transmitted utilizing Carrier Sense Multiple Access (CSMA) MAC protocol. Based on this message and to avoid hidden-terminal problem, member nodes transmit their join query messages utilizing CSMA protocol and based on the strongest received signal strength indication. Consequently, the cluster head transmits the required TDMA schedule to all member nodes using CSMA protocol. After finishing the setup phase, nodes get into the steady-state phase in which data is aggregated by the cluster head to be then immediately and directly sent to the sink node. In detail, to avoid intra-cluster interference, each node transmits its data based on the allocated TDMA time slot. Moreover, to avoid inter-cluster interference, the direct-sequence spread spectrum is used. Specifically, the TDMA schedule not only defines when member nodes need to send their data, but also a unique spreading code is associated. In other words, each cluster head employs just single match filter to differentiate its traffic, thereby avoiding inter-cluster interference. However, for the sake of having fair comparisons with prior relevant works, there is no compression employed at the cluster heads. Interestingly, in the first round of being served as a cluster

Table 4
Simulation parameters.

Parameter	Value
Network size	100 m × 100 m
Base station location	$x = 50$ m, $y = 175$ m
Number of initial nodes	100 nodes
E_{elec}	50 nJ/bit
ε_{mp}	0.0013 pJ/bit/m^4
ε_{fs}	10 nJ/bit/m^2
$h_r = h_t$	1.75 m
λ	0.328 m
Data message length	6000 bits
Control message length	900 bits
Percentage of expected cluster heads nodes	0.05
Network density	0.01 node/m^2

بداية الموت في الشبكة (First node to dead)



last node dead

Fig. 3. The number of alive nodes of the network versus number of rounds.

هو خط بعين الاعتبار انه هاي خلصت بعد ال LEACH فإذا هي أفضل بس ما حسب حساب انه First node dead بتكون بالبداية وتطول بلش ال nodes تموت .

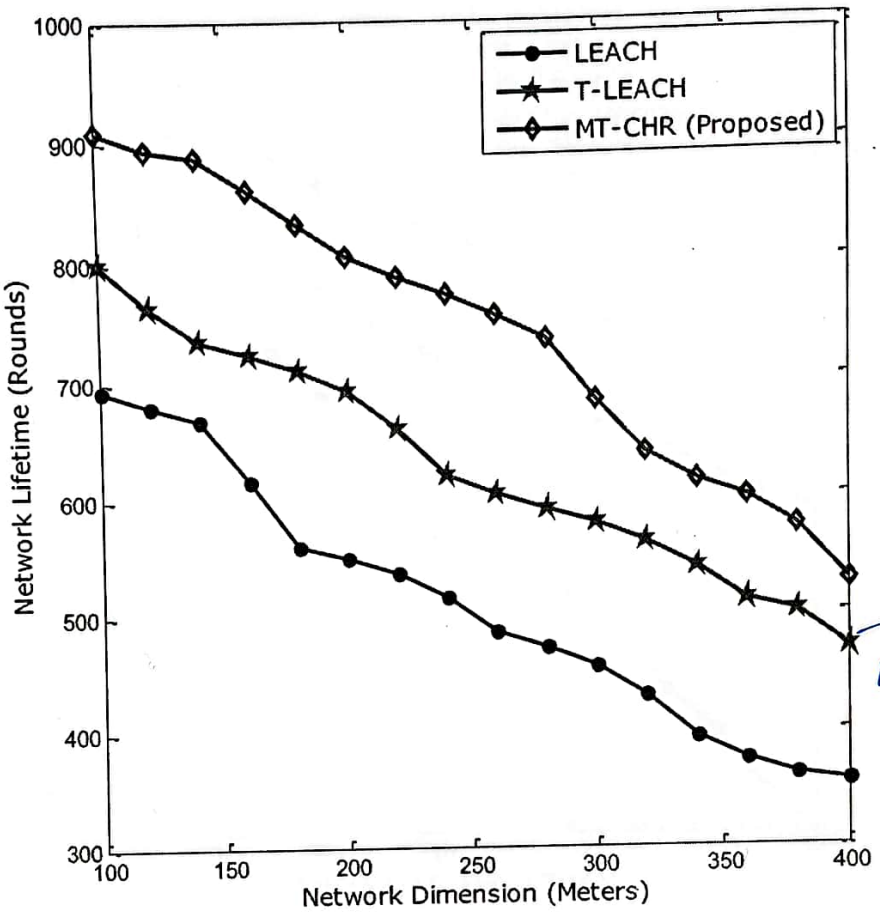
head, the threshold energy is calculated as it differs from a cluster head to another. In every round, the cluster head checks if its energy dips below that threshold. If so, the cluster head broadcasts, in the setup of next round, a disjoin query message to all network nodes, indicating that network re-clustering is required. In fact, this is done to make sure that all network nodes are timely synchronized. In every re-clustering operation, the VRC is incremented by one.

3. Simulation results

3.1. Simulation environment and parameters

LEACH, T-LEACH, and MT-CHR protocols were implemented on VB.Net 4.5 class library .dll files in which the results were saved in an SQL server 2008 R2 database. Moreover, Matlab R2013a used the .net .dll classes to print out the results, stored in the database. Simulations have been carried out on Intel Core i5 2.30 GHz and 6GB Memory on windows 8 operating sys-

for fig. 5 ← * Network Utilization
= Data Cost / (Data + Control) Cost → overhead
لو كانت ٪100 → Control overhead بكون ما في



last node dead → لانه بال LEACH كان اقصر من ال

Fig. 4. Network lifetime versus different network sizes.

tem. However, Table 4 includes all parameters used where they are assumed as default values, unless otherwise mentioned in the discussion.

3.2. Performance metrics

To analyze the performance of our protocol, the following metrics are used:

- Number of alive nodes: This metric show the number of alive nodes in the network at any specific round.
- Network lifetime: In this context, the network lifetime will be consider as the time lapsing from the start of network operation until the last node dies, measured in rounds.
- Network utilization: In this context, the network utilization refers to the ratio of data cost to the overall cost which represents both data and control overhead costs.

3.3. Results and discussion

Fig. 3 describes the number of alive nodes of the network versus number of rounds considering three different protocols, namely, LEACH, T-LEACH, as well as MT-CHR. It is interesting to notice that the number of alive nodes decreases as the number of rounds increases. This is to be expected since as the number of rounds increases, nodes' energy depletion increases which results in increasing the deaths of nodes. It is catching the attention that the deaths of T-LEACH protocol are earlier than that in LEACH protocol though it brings fascinating enhancements over LEACH protocol. This can be justified as the threshold energy, employed in T-LEACH protocol, is so low which basically results in having earlier deaths. On the contrary, the network, in T-LEACH protocol, will be functioning much longer than that in LEACH protocol. In other words, the network will be functioning in both LEACH and T-LEACH protocols for about 700 and 790 rounds, respectively. This is due to reducing control overhead in the network as there is not setup required in every round. Intriguingly, the MT-CHR protocol outperforms both LEACH and T-LEACH protocols. As can be noticed from Fig. 3, the early and late deaths, as a result of employing MT-CHR, are much beyond those in LEACH and T-LEACH. The early deaths will start at 330 rounds, while in

IET Communications

Special Issue: Future of Intelligent Wireless LANs

IET Journals
The Institution of Engineering and Technology

EDB-CHS-BOF: energy and distance-based cluster head selection with balanced objective function protocol

ISSN 1751-8628
Received on 24th January 2019
Revised 9th July 2019
Accepted on 30th July 2019
E-First on 7th October 2019
doi: 10.1049/iet-com.2019.0092
www.ietdl.org

Khalid A. Darabkh[1] ✉, *Jumana N. Zomot*[1], *Zouhair Al-qudah*[2]

[1]*Department of Computer Engineering, The University of Jordan, 11942, Amman, Jordan*
[2]*Department of Communications Engineering, Al-Hussein Bin Talal University, 20, Ma'an, Jordan*
✉ *E-mail: k.darabkeh@ju.edu.jo*

Abstract: In this research, two novel cluster head selection protocols are proposed, namely, Energy and Distance Based Cluster Head Selection (EDB-CHS) and EDB-CHS with Balanced Objective Function (EDB-CHS-BOF). They mainly aim at balancing energy consumption amongst all sensor nodes, prohibiting their precocious death, and extending the network lifetime. More specifically, in EDB-CHS protocol, a new cluster shape is proposed and consequently a tight closed-form expression for the optimal number of cluster heads (CHs) in the network is derived. Additionally, an efficacious CHS algorithm is presented, which is an expression of threshold probability that takes into account the residual energy of sensor nodes, how far they are located from the Base Station (BS), as well as the node's optimal probability of being a CH. On the other hand, the EDB-CHS-BOF protocol is proposed mainly to address the matter of having long-distance communications as a result of obtaining adjacent CHs. In EDB-CHS-BOF protocol, a new threshold probability for each sensor node to be a CH in any round is proposed. Not only to this extent, but rather a balanced objective function is proposed to ensure a well-proportioned distribution of CHs over the network. Simulation results demonstrate the superiority of the proposed protocols over other directly related works in terms of network lifetime and total data delivery.

1 Introduction

The advancements in wireless communication technologies have given rise to the development of sensors which are multifunctional smart devices of small size and low price that are capable of sensing, communicating, and conducting operations [1–4]. Environments are monitored and controlled by sensor nodes that gauge different physical phenomena like humidity, pollution levels, pressure, and so forth [5–7]. The grouping of these sensor nodes together is referred to as a wireless sensor network (WSN) [8–10].

1.1 Motivation

With the implementation of fast-evolving semiconductor technology, WSNs are proving to be an emerging powerful technology exploited in countless applications including sensing for military purposes, robotics applications, health care monitoring, environmental monitoring, and tracking of forest fires [11–13]. In fact, high-performance networks are needed to intercalate in those applications [14–16]. The main obstacles seen in WSNs, which are not prevalent in wired networks are resource constraints like finite power resources, limited communication bandwidth, and limited communication range [17–19]. This has led to the proposition of a range of routing protocols by researchers in the past few years. These protocols aim to optimally manage the limited resources at the disposal of sensor networks [20, 21].

Routing protocols in WSN are categorised into three classes: data-centric, location-based, and hierarchical [22, 23]. In hierarchical routing class, nodes are collocated in clusters, while a particular node is picked based on precise criteria to be a cluster head (CH). The CH carries on the responsibility of gathering and processing the cluster members' data, and ultimately transmitting them to the base station (BS) [23, 24]. Usually, the CH expends a plenty amount of energy in comparison to non-CH because of the additional tasks that it accomplishes. Therefore, the CH rotation is employed as a method of balancing the energy consumption within each cluster [23–25].

1.2 Related works

The most pertinent works to this work are the crucial improvements of Low Energy Adaptive Clustering Hierarchy (LEACH), which is basically considered as the father of existing hierarchical routing protocols [26]. As a major extension to LEACH protocol, the authors in [27] proposed a distributed LEACH-based CH selection strategy called LEACH with Distance-based Threshold (LEACH-DT), which is aimed at achieving energy balancing. However, sensor nodes in the LEACH-DT protocol are self-chosen to become CHs with dissimilar probabilities according to their distances to the BS. LEACH-DT protocol uses the threshold probability for CH selection, which is expressed in (1) bearing in mind that r denotes for the round number and G signifies the set of nodes that were not been CHs in recent $[1/P(s_i)]$ rounds

$$T(s_i) = \begin{cases} \dfrac{P(s_i)}{1 - P(s_i)(r \bmod [1/P(s_i)])}, & \text{if } s_i \in G \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

Specifically, LEACH-DT employs a discriminated percentage of CHs, $P(s_i)$, as a function of the distance to the BS ($d(s_i)$), which is calculated as follows [27]:

$$P(s_i) = K \frac{1/(\bar{E}_{\text{CH}}(d(s_i)) - \bar{E}_{\text{non-CH}})}{\sum_{s_j=1}^{N} 1/(\bar{E}_{\text{CH}}(d(s_j)) - \bar{E}_{\text{non-CH}})}, \quad (2)$$

where $\bar{E}_{\text{CH}}(d(s_i))$ and $\bar{E}_{\text{non-CH}}$ represent the average energy dissipation, in a single round, of a CH node and a non-CH node, respectively. One of the main drawbacks of LEACH-DT is its inability to achieve an energy balance between cluster nodes. This is due to the exclusion of the remaining energy of the nodes during the selection of CHs. Furthermore, LEACH-DT does not consider the problem of long-distance communications of adjacent CHs. To overcome some of the deficiencies of LEACH-DT, the authors in [28] suggested a Centralised Energy Efficient Distance (CEED) protocol. This protocol considers circular clusters and the

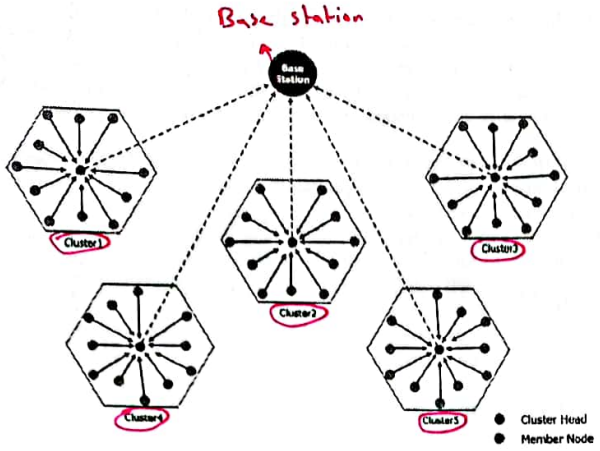


Fig. 1 *Proposed EDB-CHS and EDB-CHS-BOF network model*

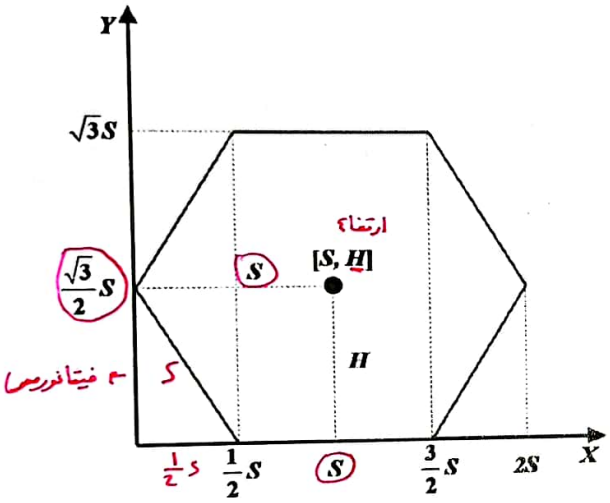


Fig. 2 *Area occupied by each cluster. The CH is centred at [S, H] of the hexagonal region*

2.3 First proposed protocol: EDB-CHS

In order to attain the maximum network lifetime in a WSN, energy consumption must be well-balanced amongst all sensor nodes. To achieve this goal, a practical or realistic cluster shape is proposed and accordingly, a closed-form expression for the optimal number of CHs is derived, bearing in mind the energy dissipation during the set-up phase as well as a steady-state phase. Based on this, a node optimal probability expression of being a CH is derived. Furthermore, a new threshold probability for CH selection is proposed. Interestingly, the new threshold probability is adaptively determined based on the node's residual energy, the distance from the sensor node to the BS, and earlier node optimal probability. The subsequent subsections thoroughly discuss the overall operation of EDB-CHS protocol, optimum number of clusters, and the proposed CH selection algorithm.

2.3.1 Overall operation of EDB-CHS: Similar to LEACH, the lifetime of EDB-CHS is divided into rounds, where a round consists of two phases: set-up and steady-state phases. The EDB-CHS protocol further divides the set-up phase into CH selection and cluster-formation algorithms. The steady-state operation is divided into frames. In every single frame, the non-CH nodes send their sensed data to their CH node during their assigned time slots in Time Division Multiple Access (TDMA) scheduling. It is worth stating that TDMA is very effective in eliminating the intra-cluster interferences. In other words, instead of letting all sensors, within a cluster, transmit their sensed data whenever they are ready and arbitrary, which will be subject to having interferences, all sensors in TDMA mechanism are given separate time slots to transmit their sensed data accordingly. Afterwards, each CH node aggregates the data and directly sending them back to the BS. Fig. 1 shows the basic network model of the EDB-CHS protocol. It is worth mentioning that most of the literature in regards to cluster-based WSNs assume, for the sake of simplicity, that the cluster area has an approximately circular shape. In this work, it is assumed that this area has a hexagonal shape since it is closer to the reality that represents the randomness in sensor nodes locations as being uniformly deployed. This can be sufficiently noticed from Fig. 1.

2.3.2 Optimal number of clusters: The number of CHs in WSNs has a considerable effect on both total energy consumption and lifetime of the network and is thus considered an important factor. When the number of clusters is low, the cluster members often must transmit sensed data vast distances in order to reach the CH, which leads to increased energy consumption. In contrast, when there are too many clusters, the amount of data that are locally aggregated will be insufficient because CHs receive data from a smaller number of members. Data collisions may also be a symptom if these clusters exist in too close of a proximity to one another, which also increases the overall energy consumption in the network.

As previously mentioned, the deployment of N sensor nodes, over an area of $M \times M$, follows a uniform distribution where K clusters amongst them are obtained which basically results in having N/K nodes per cluster (on average). In each cluster, there are $((N/K) - 1)$ non-CH nodes along with a CH. After choosing CHs, in EDB-CHS protocol, the CHs are to advertise themselves using L-bit message length each over a distance of $M/\sqrt{2}$, i.e. half-length of the network. By doing so, the communication distance is maintained below the threshold distance d_o, and thus, the use of Friss free-space propagation model is guaranteed. Consequently, the energy, required to send this advertisement (ADV) message by each CH chosen in a single round, can be expressed as

$$E_{\text{CH-ADV}} = LE_{\text{elec}} + L\varepsilon_{\text{fs}}\frac{M^2}{2}. \quad (7)$$

The energy dissipated by every non-CH node to receive the ADV message can be expressed as follows:

$$E_{\text{NON-CH-ADV}} = \frac{K}{2}LE_{\text{elec}}. \quad (8)$$

Every non-CH node chooses the cluster that it belongs to, depending on the Received Signal Strength Indicator (RSSI). Subsequently, the non-CH node sends a join-request message (Join-REQ) of length L-bit to its chosen CH. Presumably, the distance from the non-CH node to the chosen CH is somehow small, and therefore the energy dissipation follows the Friss free-space propagation (d^2 *attenuation*). Accordingly, the energy dissipated by every non-CH node to transmit the join-request (Join-REQ) message to the chosen CH can be expressed by

$$E_{\text{NON-CH-JOIN}} = LE_{\text{elec}} + L\varepsilon_{\text{fs}}d_{\text{toCH}}^2, \quad (9)$$

where d_{toCH} is the distance between the member sensor node and its CH. The area to be occupied by each cluster is approximately M^2/K and is generally an arbitrary-shaped region with a node distribution $\rho(x, y)$. The expected squared distance between the member sensor nodes and the CH node is expressed by

$$E[d_{\text{toCH}}^2] = \int\int d^2(x, y)\rho(x, y)\,dx\,dy. \quad (10)$$

Based on the prior justified cluster shape assumption, each cluster is considered to have a hexagonal shape with side length S and height $2H$, as shown in Fig. 2. Consequently, the distance from the centre of the hexagon (S, H) to the midpoint of a side can be defined as $H = S\sqrt{3}/2$. Therefore, the following can be obtained:

IET Commun., 2019, Vol. 13 Iss. 19, pp. 3168-3180

$$\frac{M^2}{K} = \frac{3\sqrt{3}}{2}S^2. \quad (11)$$

Hence

$$S = \sqrt{\frac{4}{27}}\frac{M}{\sqrt{K}}. \quad (12)$$

Additionally, the CH node is assumed to be situated at the centre of the cluster, as well as $\rho(x, y)$ is constant for x and y.

Consequently, (10) can be articulated as follows:

$$E[d_{toCH}^2] = \rho\int_{x=0}^{x=(1/2)S}\int_{y=(\sqrt{3}/2)S-\sqrt{3}x}^{y=\sqrt{3}x+(\sqrt{3}/2)S}(x-S)^2+(y-H)^2\,dx\,dy + \int_{x=S/2}^{x=(3/2)S}\int_{y=0}^{y=\sqrt{3}S}(x-S)^2+(y-H)^2\,dx\,dy + \int_{x=(3/2)S}^{x=2S}\int_{y=\sqrt{3}x-(3\sqrt{3}/2)S}^{y=(5\sqrt{3}/2)S-\sqrt{3}x}(x-S)^2+(y-H)^2\,dx\,dy. \quad (13)$$

By evaluating the integrals and then substituting the value of H, the following is obtained:

$$E[d_{toCH}^2] = \rho\frac{\sqrt{3}}{2}S^4. \quad (14)$$

Substituting (12) into (14) yields

$$E[d_{toCH}^2] = \rho\frac{\sqrt{12}\,M^4}{27\,K^2}. \quad (15)$$

If the density of sensor nodes is uniform all through the cluster area, then $\rho = 1/(M^2/K)$ and

$$E[d_{toCH}^2] = \frac{\sqrt{12}\,M^2}{27\,K}. \quad (16)$$

Complete details about deriving (16) are provided in the Appendix.

Therefore, in this case (9) can be written as

$$E_{NON-CH-JOIN} = LE_{elec} + L\varepsilon_{fs}\frac{\sqrt{12}\,M^2}{27\,K}. \quad (17)$$

The energy dissipated by every CH node to receive Join-REQ message from the cluster members can be expressed as follows:

$$E_{CH-JOIN} = \left(\frac{N}{K}-1\right)LE_{elec}. \quad (18)$$

Each CH node creates a TDMA schedule of length L-bit and transmits it to every sensor node in its cluster. Therefore, the energy dissipated by every CH is

$$E_{CH-TDMA} = LE_{elec} + L\varepsilon_{fs}\frac{\sqrt{12}\,M^2}{27\,K}. \quad (19)$$

The energy dissipated by every non-CH node to receive a TDMA schedule can be expressed by

$$E_{NON-CH-TDMA} = LE_{elec}. \quad (20)$$

The set-up phase is complete as soon as non-cluster nodes receive their TDMA schedule, and then the steady-state phase (data transmission) can begin. In this phase, every non-CH node sends its sensed data of length L_{DATA}-bit to its respective CH node during its assigned transmission slot. Thus, the energy dissipated by every non-cluster node to transmit its data is given by

$$E_{NON-CH-DATA} = L_{DATA}E_{elec} + L_{DATA}\varepsilon_{fs}\frac{\sqrt{12}\,M^2}{27\,K}. \quad (21)$$

If sensor nodes neither hear the advertisement messages from the CHs nor the Join-REQ messages from non-CH nodes, they assume that no CHs exist in the network. Accordingly, every sensor node transmits its sensed data to the BS individually. In other words, the sensor nodes fail to find their CH. However, after the CH receives data from its cluster members, it aggregates the data and consequently transmits it to the BS. Therefore, the energy used by CH during the steady-state phase can be expressed by

$$E_{CH-DATA} = \left(\frac{N}{K}-1\right)L_{DATA}E_{elec} + \frac{N}{K}L_{DATA}E_{DA} + L_{DATA}E_{elec} + L_{DATA}\varepsilon_{mp}d_{toBS}^4, \quad (22)$$

where E_{DA} symbolises the energy expended for data aggregation, and d_{toBS} signifies the distance from the CH node to the BS. Thus, the total energy dissipated by every CH node in a single round can be defined as follows:

$$E_{CH} = E_{CH-ADV} + E_{CH-JOIN} + E_{CH-TDMA} + E_{CH-DATA}. \quad (23)$$

Substituting the values of $E_{CH\text{-}ADV}$, $E_{CH\text{-}JOIN}$, $E_{CH\text{-}TDMA}$, and $E_{CH\text{-}DATA}$ from (7), (18), (19) and (22) into (23) yields

$$E_{CH} = 2LE_{elec} + L\varepsilon_{fs}\frac{M^2}{2} + \left(\frac{N}{K}-1\right)LE_{elec} + L\varepsilon_{fs}\frac{\sqrt{12}\,M^2}{27\,K} + \left(\frac{N}{K}-1\right)L_{DATA}E_{elec} + \frac{N}{K}L_{DATA}E_{DA} + L_{DATA}E_{elec} + L_{DATA}\varepsilon_{mp}d_{toBS}^4. \quad (24)$$

The total energy dissipated by every non-CH node in a single round is expressed as follows:

$$E_{NON-CH} = E_{NON-CH-ADV} + E_{NON-CH-JOIN} + E_{NON-CH-TDMA} + E_{NON-CH-DATA}. \quad (25)$$

Substituting values of $E_{NON\text{-}CH\text{-}ADV}$, $E_{NON\text{-}CH\text{-}JOIN}$, $E_{NON\text{-}CH\text{-}TDMA}$, and $E_{NON\text{-}CH\text{-}DATA}$ from (8), (17), (20) and (21) into (25) gives

$$E_{NON-CH} = \frac{K}{2}LE_{elec} + 2LE_{elec} + L\varepsilon_{fs}\frac{\sqrt{12}\,M^2}{27\,K} + L_{DATA}E_{elec} + L_{DATA}\varepsilon_{fs}\frac{\sqrt{12}\,M^2}{27\,K}. \quad (26)$$

Consequently, the overall energy dissipation within the network, in every round, can be defined as follows:

$$E_{TOTAL-NW} = KE_{CLUSTER}. \quad (27)$$

where $E_{CLUSTER}$ signifies to the total energy dissipated within each cluster, and is expressed as follows:

$$E_{CLUSTER} = E_{CH} + \left(\frac{N}{K}-1\right)E_{NON-CH}. \quad (28)$$

Applying that $((N/K)-1) \simeq N/K$ and accordingly (28) provides

$$E_{CLUSTER} = E_{CH} + \frac{N}{K}E_{NON-CH}. \quad (29)$$

Substituting (24) and (26) into (29) and consequently into (27) and after rearranging, the following is obtained:

$$E_{\text{TOTAL-NW}} = \left(2K + 3N + N\frac{K}{2}\right)LE_{\text{elec}} + KL\varepsilon_{\text{fs}}\frac{M^2}{2} + (K+N)L\varepsilon_{\text{fs}}\frac{\sqrt{12}}{27}\frac{M^2}{K} + (2N+K)L_{\text{DATA}}E_{\text{elec}} + NL_{\text{DATA}}E_{\text{DA}} + KL_{\text{DATA}}\varepsilon_{mp}d_{\text{toBS}}^4 + NL_{\text{DATA}}\varepsilon_{\text{fs}}\frac{\sqrt{12}}{27}\frac{M^2}{K}. \quad (30)$$

Interestingly, the optimal number of clusters can be calculated by taking the derivative of $E_{\text{TOTAL-NW}}$ in (30) with respect to K, and set the result of the derivative to zero as follows:

$$\frac{\partial E_{\text{TOTAL-NW}}}{\partial K} = 0, \quad (31)$$

Consequently, the optimal number of clusters is calculated as shown in the following equation: (see (32)) . In EDB-CHS, the optimal probability of any sensor node s_i to become a CH can be expressed as follows:

$$P_{\text{OPT}}(s_i) = \frac{K_{\text{OPT}}}{N}. \quad (33)$$

$P_{\text{OPT}}(s_i)$ is chosen in such a manner that the expected number of CHs for the current round is K_{OPT}. Accordingly, the following constraint can be shown:

$$\sum_{s_i=1}^{N} P_{\text{OPT}}(s_i) = K_{\text{OPT}}. \quad (34)$$

2.3.3 EDB-CHS: CH selection algorithm: In EDB-CHS protocol, a new threshold probability for CH selection is proposed in a manner that ensures a balanced energy consumption amongst all sensor nodes. However, in this threshold probability, a load of the residual energy as well as the average residual energy of sensor nodes are gotten to make sure that the sensor node with higher residual energy and less energy consumption has a higher chance to be a CH. Furthermore, the distance between the sensor node and BS has been incorporated into the threshold probability, thereby ensuring that the nearest sensor node has more opportunity to be chosen as the CH than the farthest sensor node, which can lead to substantial energy saving and prolonging the network lifetime in terms of the last node to die (LND) metric that will be discussed in a little while. Not only to this extent but rather it has taken advantage of an earlier derived optimal number of CHs to derive a new node optimal probability of being a CH to be then incorporated into that threshold probability. Therefore, the new threshold $P_T(s_i)$ for the sensor node s_i is suggested to be as given below: (see (35)) , where $P_{\text{OPT}}(s_i)$ signifies the optimal probability of a sensor node s_i to qualify as a CH node in the present round, $E_{\text{CUR}}(s_i)$ represents the residual energy of s_i sensor node, E_{AVG} symbolises the average residual energy for sensor nodes, $d_{\text{AVG-toBS}}$ implies the average distance between sensor nodes and BS, and $d_{\text{toBS}}(s_i)$ implies the distance from the sensor node s_i to BS. G signifies the set of nodes which have never been CHs in the last $[1/P_{\text{OPT}}(s_i)]$ rounds, where

$$E_{\text{AVG}} = \frac{\sum_{s_i=1}^{N} E_{\text{CUR}}(s_i)}{N}, \quad (36)$$

and

$$d_{\text{AVG-toBS}} = \frac{\sum_{s_i=1}^{N} d_{\text{toBS}}(s_i)}{N}. \quad (37)$$

At the beginning of the set-up phase, sensor nodes find the threshold probability $P_T(s_i)$, according to the formula introduced in (35). Therefore, every sensor node s_i chooses a random number within the range [0, 1]. If the random number obtained is less than $P_T(s_i)$, then that sensor node elects itself as a CH. Otherwise, it will become one of the cluster members. After selecting the CHs, each CH broadcasts an ADV message to all nearby sensor nodes using Carrier Sense Multiple Access (CSMA) MAC protocol. As the clusters have not been formed at this time, the use of TDMA mechanism will not be possible. Therefore, to eliminate the possible interferences between sensors, CSMA becomes of a great interest, i.e. each sensor senses the channel before transmitting its sensed data. However, the ADV message contains the sensor node's identifier (ID) as well as a header that discriminates this message as a declaration message. Based on the RSSI of this short message, each sensor node transmits its join-request message to the closest CH using the CSMA protocol. After the clusters are formed, each CH produces a TDMA schedule and sends it to all its cluster members. This averts collision between data messages and minimises the dissipated energy for each non-CH node by permitting its radio components to be turned off all times except in its transmit time. Moreover, the direct-sequence spread spectrum is employed for the reason of reducing the inter-cluster interference, wherein each cluster uses a unique spreading code. Specifically, all cluster members send their data using this spreading code that is enclosed in the TDMA schedule sent earlier and therefore the CH can easily filter all received signals.

2.4 Second proposed protocol: EDB-CHS-BOF

As previously discussed, having adjacent CHs may cause long distance communications, which lead to increased energy consumption. To solve this problem, the second protocol is proposed, i.e. EDB-CHS-BOF. In EDB-CHS-BOF protocol, a distributed CH selection algorithm is introduced in which sensor nodes are chosen to become CHs based on a derived closed-form expression that refers to the node optimal probability. Therefore, an amendment on the CH selection threshold probability, proposed in EDB-CHS protocol, is considered through including the expression of node optimal probability. Moreover, it has been amended through inserting the parameter of optimal number of CHs for the reason of increasing the number of candidate CHs and then setting constraints on selecting appropriate CHs for ultimately achieving a long lasting network lifetime. These improvements will be discussed in further details in the next subsection.

2.4.1 EDB-CHS-BOF: CH selection algorithm: Unlike that proposed in EDB-CHS protocol, in EDB-CHS-BOF protocol, the optimal probability of a sensor node s_i to be a CH node is derived based on a balanced formula for the energy consumption and how distant nodes are from BS. Hence, it becomes variable and depends on these dimensions (i.e. nodes' energy and distances to the BS). Nevertheless, like EDB-CHS, $P_{\text{OPT}}(s_i)$ is chosen in such a manner

$$K_{\text{OPT}} = M\frac{\sqrt{12}}{\sqrt{27}}\sqrt{\frac{(L+L_{\text{DATA}})N\varepsilon_{\text{fs}}}{L(2E_{\text{elec}} + (N/2)E_{\text{elec}} + \varepsilon_{\text{fs}}(M^2/2)) + L_{\text{DATA}}(E_{\text{elec}} + \varepsilon_{mp}d_{\text{toBS}}^4)}} \quad (32)$$

$$P_T(s_i) = \begin{cases} \dfrac{P_{\text{OPT}}(s_i)}{1 - P_{\text{OPT}}(s_i)(r \bmod [1/P_{\text{OPT}}(s_i)])} \cdot \dfrac{E_{\text{CUR}}(s_i)}{E_{\text{AVG}}} \cdot \dfrac{d_{\text{AVG-toBS}}}{d_{\text{toBS}}(s_i)}, & \text{if } s_i \in G \\ 0, & \text{otherwise} \end{cases} \quad (35)$$

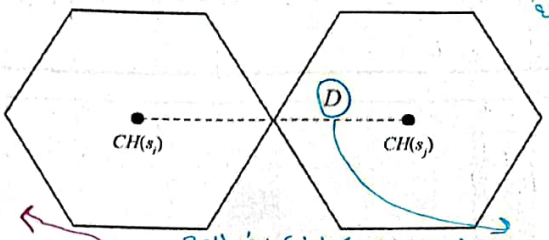


Fig. 3 *Average distance between two adjacent CHs*

that the expected number of CHs for the current round is K_{OPT}, which is expressed, as shown in (34).

Different from EDB-CHS protocol, if it is assumed that nodes s_i and s_j are taking their turns as CHs with probabilities $P_{OPT}(s_i)$ and $P_{OPT}(s_j)$, respectively, and the distances of nodes s_i and s_j to the BS are $d(s_i)$ and $d(s_j)$, respectively, then the following balance formula for the energy consumption is obtained:

$$P_{OPT}(s_i)E_{CH}(s_i)d(s_i) + (1 - P_{OPT}(s_i))E_{NON-CH}\sqrt{\frac{\sqrt[4]{12}}{27}}\sqrt{\frac{M}{K_{OPT}}} = P_{OPT}(s_j)E_{CH}(s_j)d(s_j) + (1 - P_{OPT}(s_j))E_{NON-CH}\sqrt{\frac{\sqrt[4]{12}}{27}}\sqrt{\frac{M}{K_{OPT}}} \quad (38)$$

After simplifying, the following is ultimately obtained:

$$P_{OPT}(s_i) = \frac{P_{OPT}(s_j)\left(E_{CH}(s_j)d(s_j) - E_{NON-CH}\sqrt{\frac{\sqrt[4]{12}}{27}}\sqrt{\frac{M}{K_{OPT}}}\right)}{E_{CH}(s_i)d(s_i) - E_{NON-CH}\sqrt{\frac{\sqrt[4]{12}}{27}}\sqrt{\frac{M}{K_{OPT}}}} \quad (39)$$

Assuming that

$$u = P_{OPT}(s_j)\left(E_{CH}(s_j)d(s_j) - E_{NON-CH}\sqrt{\frac{\sqrt[4]{12}}{27}}\sqrt{\frac{M}{K_{OPT}}}\right), \quad (40)$$

and

$$\psi(s_i) = \frac{1}{E_{CH}(s_i)d(s_i) - E_{NON-CH}\sqrt{\frac{\sqrt[4]{12}}{27}}\sqrt{\frac{M}{K_{OPT}}}}. \quad (41)$$

Then, it is obtained $P_{OPT}(s_i) = u\psi(s_i)$. Using (34), it can be found that $u = K_{OPT}/\sum_{s_j=1}^{N}\psi(s_j)$ and therefore $P_{OPT}(s_i)$ can be written as the following equation:

$$P_{OPT}(s_i) = K_{OPT}\frac{\psi(s_i)}{\sum_{s_j=1}^{N}\psi(s_j)}, \quad 0 \le P_{OPT}(s_i) \le 1. \quad (42)$$

It is noteworthy to mention that it has been noticed through enormous simulations that EDB-CHS protocols may result in having adjacent (very close) CHs. Therefore, EDB-CHS-BOF protocol adds to the threshold probability expression derived in EDB-CHS protocol a new parameter which refers to the optimal number of CHs and this is for the sake of increasing the number of candidate CHs in a reasonable way which is then to be filtered using the balanced objective function that will be discussed shortly. Thus, the new threshold probability $P_H(s_i)$ becomes as given below: (see (43)) . In this equation, K_{OPT} symbolises the optimum number of CHs which is obviously alluded in (32). Other details concerning the generation of random numbers by nodes that are then compared with just the elaborated threshold probability for possible CH section remains as proposed in EDB-CHS protocol. Nevertheless, to make sure that the CHs in any round are not closely located, a balanced objective function that basically ensures a well-proportioned distribution of CHs over the network is introduced. Therefore, the distance that each candidate CH examines to see if it is time to declare itself as a CH node for the current round is expressed by the following balanced objective function:

$$D_{s_j} = \arg\min \| CCH(s_j) - CH(s_i) \|^2, \quad \forall i = 1, 2, \ldots, n \quad (44)$$

where $\arg\min \| CCH(s_j) - CH(s_i) \|^2$ is the Euclidean distance between the candidate cluster head s_j and the prior selected cluster heads s_i. Impressively, as shown in Fig. 3, the average distance between two adjacent CHs, $CH(s_i)$ and $CH(s_j)$, denoted by D, is proposed to be equal to the twice of the expected distance between any given CH and its cluster members. Therefore, the threshold distance is expressed as follows:

$$TH_D = 2E[d_{toCH}] = 2\sqrt{\frac{\sqrt[4]{12}}{27}}\sqrt{\frac{M}{K_{OPT}}}. \quad (45)$$

If the objective function is greater than TH_D threshold distance, then the candidate CH chooses itself as a CH node for the current round, and hence, broadcasts advertisement message to all nearby sensor nodes. Otherwise, the candidate CH becomes a non-CH node, and accordingly chooses which cluster it should join in the same manner that is explained earlier. It is worth noting that the candidate CH can estimate its distance from the prior selected CHs in accordance with the RSSI of advertisement messages.

3 Simulation results

3.1 Simulation environment and parameters

To evaluate the performance of EDB-CHS and EDB-CHS-BOF protocols, simulations have been run using MATLAB. Moreover, the simulations have been executed on a computer device with Intel core i7 processor working at 1.60 GHz with 16 GB RAM, and runs Windows 10 Enterprise edition (64 bit). In the simulation experiments, 100 sensor nodes are uniformly deployed over (200 m × 200 m). The BS is located at coordinates (100 m × 300 m). However, Table 1 summarises the main parameters employed in the simulation.

3.2 Performance metrics

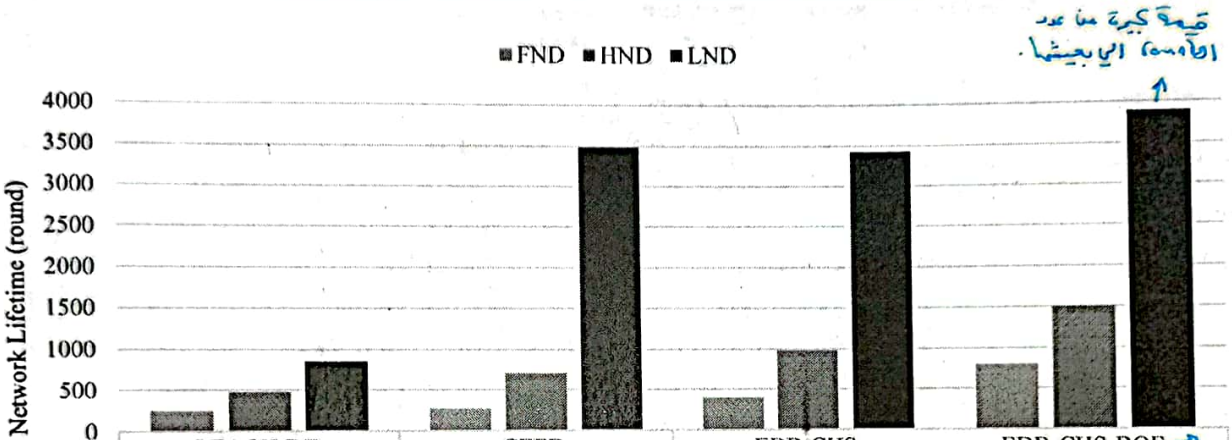
Several simulation experiments were performed to evaluate the efficiency of the proposed protocols, whereas the following performance metric is used:

$$P_H(s_i) = \begin{cases} \frac{P_{OPT}(s_i)}{1 - P_{OPT}(s_i)(r \bmod [1/P_{OPT}(s_i)])} \cdot \frac{E_{CUR}(s_i)}{E_{AVG}} \cdot \frac{d_{AVG-toBS}}{d_{toBS}(s_i)} \cdot K_{OPT}, & \text{if } s_i \in G \\ 0, & \text{otherwise} \end{cases} \quad (43)$$

* سؤال امتحان: بقللو ليه قيمة المعادلة انما ما تعتمد ع (Kopt) ؟
* سؤال امتحان: ليه رجعنا عملنا الـ objective funct. (معادلة 45) ؟
* سؤال امتحان: بعطيك Kopt وسيناريو معين وبطلب تحسب $P_H(Si)$ ؟

Table 1 Simulation parameters

Parameter	Values
network size ($M \times M$)	200 m × 200 m
BS location (X, Y)	100 m, 300 m
number of sensor nodes (N)	100 nodes
initial energy (E_o)	0.5 J
radio electronics energy (E_{elec})	50 nJ/bit
transmit amplifier energy (ε_{fs})	10 pJ/bit/m^2
transmit amplifier energy (ε_{mp})	0.0013 pJ/bit/m^4
cross-over distance (d_o)	87 m
data aggregation energy (E_{DA})	5 nJ/bit/signal
length of data message (L_{DATA})	500 bits
length of packet header (L)	64 bits



قيمة كبيرة منا عدد الأمة اللي بعيشها.

	LEACH-DT	CEED	EDB-CHS	EDB-CHS-BOF
FND	247	269	398	798
HND	480	707	986	1506
LND	854	3490	3430	3893

اول node بموت ← | نص ← | اخر node ←

اللي فيه balance

Fig. 4 *Network lifetime for all protocols in a sensor field of size 200 × 200 m^2*

- Network Lifetime which primarily includes First Node to Die (FND), Last Node to Die (LND), in addition to the Half Node to Die (HND).
- Total data delivery which represents the total amount of data messages received by the BS considering FND, HND, as well as LND network lifetime indicators.

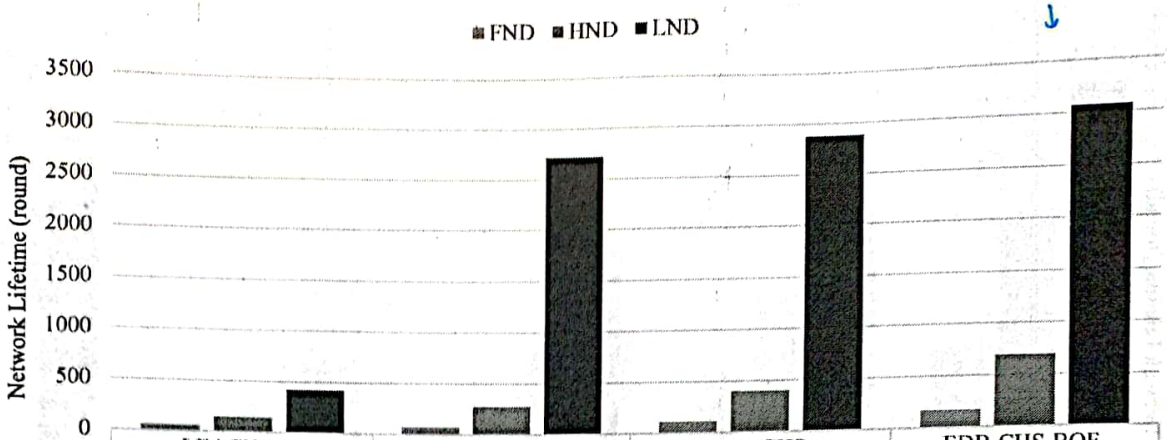
3.3 Results and discussion

In this section, the performance of the proposed protocols (EDB-CHS and EDB-CHS-BOF) is investigated in terms of the three different descriptions of the network lifetime. Furthermore, two of the most relevant works, LEACH-DT and CEED, are implemented, and subsequently, fair comparisons are conducted between the results of the proposed protocols and those attained from these prior protocols. Finally, the influence of increasing network size on the performance of EDB-CHS, EDB-CHS-BOF, and related works is examined.

3.3.1 Network lifetime investigations: The examination of network lifetime in terms of the FND, HND, and LND metrics is intensively studied in this subsection and clearly depicted in Fig. 4 for the proposed protocols as well as the LEACH-DT and CEED protocols. The simulation was run 20 times, and the mean of these simulations was considered to obtain an accurate network lifetime.

It can be clearly seen from Fig. 4 that EDB-CHS protocol beats both LEACH-DT and CEED protocols from the points of view of the FND, HND, and LND measurements. Particularly, the FND metric achieved in EDB-CHS protocol is 398 rounds, whereas, in LEACH-DT and CEED protocols, they are 247 and 269 rounds, respectively, which basically results in having enhancements of 61 and 48% over LEACH-DT and CEED protocols, respectively. As far as the HND metric is concerned, its values are 986, 480, and 707 rounds for EDB-CHS, LEACH-DT, and CEED protocols, respectively. Amazingly, this provides improvements of EDB-CHS protocol over LEACH-DT and CEED protocols in percentages of 105 and 39%, respectively. Concerning LND metric, its values are 3430, 854 and, 3490 rounds for EDB-CHS, LEACH-DT, and CEED protocols, respectively. This results in having a colossal change of 302% over LEACH-DT, while CEED achieved a minor improvement of 2% over EDB-CHS.

These tremendous results are related directly to the main improvements introduced in EDB-CHS protocol. Firstly, deriving a tight closed-form expression for the optimal number of CHs, based on a practical cluster shape, has a positive influence on reducing the energy consumption and henceforth prolonging the network lifetime. As a result, a new optimal probability of a node to be a CH is derived. Secondly, EDB-CHS protocol employs an efficient method of selecting the CHs which basically involves a new threshold probability that takes into account the residual energy of sensor node and its distance to BS along with node's optimal probability. Thirdly, the CHs in EDB-CHS protocol broadcast the advertisement messages to a distance of half-length of the network instead of the entire network, as was proposed indirectly connected works. This means that the CHs will usually use the Friss free-space propagation model (d^2 *attenuation*) rather than the two-ray propagation model (d^4 *attenuation*) which makes the energy depletion lesser. Finally, EDB-CHS limits the distance over which the TDMA schedules are broadcasted. Specifically, the CHs compute the maximum distance to reach their cluster members and accordingly send the TDMA schedule over that distance, thereby decreasing the energy consumption within the entire network. It



	LEACH-DT	CEED	EDB-CHS	EDB-CHS-BOF
FND	57	72	110	170
HND	143	275	399	690
LND	418	2716	2878	3066

Fig. 5 *Network lifetime for all protocols in a sensor field of size 300 × 300 m^2*

can be further observed from Fig. 4 that CEED protocol has a better performance than LEACH-DT protocol as it considers the sensor node energy as well as its location when picking the CHs. On the other hand, CEED employs the optimal number of CHs rather than the expectation as used in LEACH-DT.

To this end, it is completely clear that EDB-CHS-BOF protocol is the best performer in terms of aforementioned definitions of the network lifetime (i.e. FND, HND, and LND metrics). Precisely, the FND metric of EDB-CHS-BOF is at 798 rounds, which leads to having remarkable enhancements of 101, 223, and 197% over EDB-CHS, LEACH-DT, and CEED protocols, respectively. The HND metric for EDB-CHS-BOF is at 1506 rounds. This results in having enhancements of 53, 214, and 113% over EDB-CHS, LEACH-DT, and CEED protocols, respectively. Finally, the LND metric of EDB-CHS-BOF is at 3893 rounds. This results in having superior improvement of 356% over LEACH-DT and small improvements of 14 and 12% over EDB-CHS and CEED, respectively.

Actually, these enhancements were achieved by employing the following ideas. At first, EDB-CHS-BOF protocol adopts a distributed CH selection algorithm that ends up in deriving a threshold probability which takes into account the sensor node energy dissipation in each round, its distance to BS, the optimal number of CHs, as well as the optimal probability of a node to serve as a CH that is derived utilising a balanced formula of energy consumption and how distant nodes are from the BS. This in turn balances energy consumption amongst the sensor nodes, which certainly makes the sensor network remain functioning for a longer period. Moreover, further, improvement is introduced in the CH selection algorithm by primarily guaranteeing effective CHs distribution throughout the network and avoids the adjacent formation of clusters. This results in a reduction of the number of long distance communications with BS and succeeds in achieving a longer lifespan of the whole network.

3.3.2 Network lifetime for different network sizes: In this subsection, the influence of increasing the network size on the performance of the proposed protocols along with directly connected works, in terms of FND, HND, and LND metrics is examined. In the simulations conducted, two different network sizes are considered: (300 m × 300 m) and (400 m × 400 m). The locations of the BS for these network sizes are (150 m × 400 m) and (200 m × 500 m), respectively. It is clearly shown from Figs. 5 and 6 that all protocols incline to have a shorter lifetime with the increases of network size. This is due to communications over long distances which cause the sensor nodes to die faster.

LEACH-DT has the worst performance where the first node dies at rounds 57 and 17 and last node dies at rounds 418 and 152 when network sizes are 300 m × 300 m and 400 m × 400 m, respectively. Actually, this is expected since LEACT-DT does not bear in mind nodes' residual energy when selecting the CHs, which surely upsurges the number of dead nodes, influencing the network lifetime.

For all network sizes, EDB-CHS protocol has a better performance than CEED with respect to the FND and HND metrics. Referring to Figs. 5 and 6, EDB-CHS protocol shows improvements in terms of FND of 53 and 133% over CEED when network sizes are 300 m × 300 and 400 m × 400 m, respectively. It also shows improvements in terms of HND of 45 and 73% over CEED when network sizes are 300 m × 300 m and 400 m × 400 m, respectively. This is due to the enhancements implemented in EDB-CHS protocol, which are extensively discussed and highlighted in the previous section. On the other hand, EDB-CHS protocol has a slightly higher performance than CEED with respect to the LND metric, for all network sizes. This can be related to the fact that EDB-CHS and CEED consider the residual energy of nodes besides the distance to BS in the CH selection process. Furthermore, as previously discussed, CEED reduces the possibility of a sensor node becoming a CH, which leads to the existence of rounds where no clusters are formed. Therefore, each sensor node transmits its sensed data individually to the BS.

It can be further seen that EDB-CHS-BOF protocol has the best performance with respect to the FND, HND, and LND metrics, for all network sizes. Specifically, EDB-CHS-BOF improves the FND by 55, 136, and 198%, HND by 73, 151, and 383%, as well as LND by 7, 13, and 633% over EDB-CHS, CEED, and LEACH-DT, respectively, when the network size equal to 300 m × 300 m. EDB-CHS-BOF improves the FND by 7, 150, and 165%, HND by 45, 150, and 342%, as well as LND by 38, 58, and 1869% over EDB-CHS, CEED, and LEACH-DT, respectively, when the network size equal to 400 m × 400 m. Actually, the amendments introduced in EDB-CHS-BOF protocol, which unquestionably ensures a good distribution of the CHs in the network, contribute significantly in extending the network lifetime and making its performance preferable over all other protocols.

3.3.3 Total data delivery for different network sizes and network lifetime indicators: To this extent, it will be interesting to study how the performance of the proposed protocols acts in the perspectives of total data delivery considering three network lifetime indicators (i.e. FND, HND, and LND), three network sizes (i.e. 200 m × 200 m, 300 m × 300 m, as well as 400 m × 400 m), and the exclusion of data aggregation. The results of just mentioned performance metrics for the proposed protocols and directly connected ones are shown in Figs. 7–9, respectively.

[15] Darabkh, K.A., Ismail, S.S., Al-Shurman, M., *et al.*: 'Performance evaluation of selective and adaptive heads clustering algorithms over wireless sensor networks', *J. Netw. Comput. Appl.*, 2012, **35**, (6), pp. 2068–2080

[16] Ismail, S.S., Al Khader, A.I., Darabkh, K.A.: 'Static clustering for target tracking in wireless sensor networks', *Global J. Technol. (Selected Paper of COMENG-2014)*, 2015, **8**, pp. 167–173

[17] Darabkh, K.A., El-Yabroudi, M.Z., El-Mousa, A.H.: 'BPA-CRP: A balanced power-aware clustering and routing protocol for wireless sensor networks', *Ad Hoc Netw.*, 2019, **82**, pp. 155–171

[18] Darabkh, K.A., Judeh, M.S.E.: 'An Improved Reactive Routing Protocol over Mobile Ad-hoc Network', Proceedings of the 14th IEEE International Wireless Communications and Mobile Computing Conference (IWCMC 2018), Limassol, Cyprus, June 2018, pp. 707–711

[19] Khalifeh, A.F., AlQudah, M., Darabkh, K.A.: 'Optimizing the beacon and super frame orders in IEEE 802.15.4 for real-time notification in wireless sensor networks'. Int. Conf. on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, India, March 2017, pp. 595–598

[20] Darabkh, K.A., Al-Maaitah, N.J., Jafar, I.F., *et al.*: 'EA-CRP: a novel energy-aware clustering and routing protocol in wireless sensor networks', *Comput. Electr. Eng.*, 2018, **72**, pp. 702–718

[21] Darabkh, K.A., Al-Maaitah, N.J., Jafar, I.F., *et al.*: 'Energy efficient clustering algorithm for wireless sensor networks'. Int. Conf. on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, India, March 2017, pp. 590–594

[22] Darabkh, K.A., Albtoush, W.Y., Jafar, I.F.: 'Improved clustering algorithms for target tracking in wireless sensor networks', *J. Supercomput.*, 2017, **73**, (5), pp. 1952–1977

[23] Darabkh, K.A., Al-Rawashdeh, W.S., Hawa, M., *et al.*: 'MT-CHR: A modified threshold-based cluster head replacement protocol for wireless sensor networks', *Comput. Electr. Eng.*, 2018, **72**, pp. 926–938

[24] Darabkh, K.A., Muqat, R.Z.: 'An efficient protocol for minimizing long-distance communications over wireless sensor networks'. Int. Multi-Conf. on Systems, Signals & Devices (SSD), Hammamet, Tunisia, March 2018, pp. 671–676

[25] Darabkh, K.A., Alsaraireh, N.R.: 'A yet efficient target tracking algorithm in wireless sensor networks'. Int. Multi-Conf. on Systems, Signals & Devices (SSD), Hammamet, Tunisia, March 2018, pp. 7–11

[26] Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: 'Energy-efficient communication protocol for wireless microsensor networks'. Proc. of the 33rd Annual Hawaii Int. Conf. on System Sciences, Maui, HI, USA, January 2000, p. 10

[27] Kang, S.H., Nguyen, T.: 'Distance based thresholds for cluster head selection in wireless sensor networks', *IEEE Commun. Lett.*, 2012, **16**, (9), pp. 1396–1399

[28] Gawade, R.D., Nalbalwar, S.L.: 'A centralized energy efficient distance based routing protocol for wireless sensor networks', *J. Sens.*, 2016, **2016**, Article ID 8313986, p. 8

[29] Darabkh, K.A., Zomot, J.N.: 'An improved cluster head selection algorithm for wireless sensor networks'. 2018 14th Int. Wireless Communications & Mobile Computing Conf. (IWCMC), Limassol, Cyprus, June 2018, pp. 65–70

[30] Darabkh, K.A., Odetallah, S.M., Al-qudah, Z., *et al.*: 'Energy–aware and density-based clustering and relaying protocol (EA-DB-CRP) for gathering data in wireless sensor networks', *Appl. Soft Comput.*, 2019, **80**, pp. 154–166

[31] Darabkh, K.A., Al-Jdayeh, L.: 'AEA-FCP: an adaptive energy-aware fixed clustering protocol for data dissemination in wireless sensor networks', *Pers. Ubiquitous Comput.*, doi: https://doi.org/10.1007/s00779-019-01233-0

[32] Al-Mistarihi, M.F., Tanash, I.M., Yaseen, F.S., *et al.*: 'Protecting source location privacy in a clustered wireless sensor networks against local eavesdroppers', *Mobile Netw. Appl.*, doi: https://doi.org/10.1007/s11036-018-1189-6

[33] Darabkh, K.A., Al-Rawashdeh, W.S., Al-Zubi, R.T., *et al.*: 'C-DTB-CHR: centralized density-and threshold-based cluster head replacement protocols for wireless sensor networks', *J. Supercomput.*, 2017, **73**, (12), pp. 5332–5353

[34] Darabkh, K.A., Al-Rawashdeh, W.S., Al-Zubi, R.T., *et al.*: 'A new cluster head replacement protocol for wireless sensor networks'. 2017 European Conf. on Electrical Engineering and Computer Science (EECS), Bern, Switzerland, November 2017, pp. 472–476

[35] Darabkh, K.A., El-Yabroudi, M.Z.: 'A reliable relaying protocol in wireless sensor networks'. 2017 European Conf. on Electrical Engineering and Computer Science (EECS), Bern, Switzerland, November 2017, pp. 56–60

[36] Al-Zubi, R.T., Abedsalam, N., Atieh, A., *et al.*: 'LBCH: load balancing cluster head protocol for wireless sensor networks', *Informatica*, 2018, **29**, (4), pp. 633–650

6 Appendix

6.1 Expected squared distance calculation (derivation of (16))

The expected squared distance between member sensor nodes and the CH node $E[d_{toCH}^2]$, (16), is extensively derived in this Appendix. The regular hexagon is composed of six equilateral triangles, as shown in Fig. 10. Therefore, the area of a hexagon is calculated as follows:

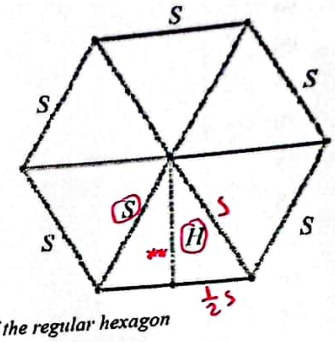


Fig. 10 *Area of the regular hexagon*

$$A_{hexagon} = 6 \times A_{traingle} = 6 \times \frac{1}{2} \times S \times H. \tag{46}$$

In this equation, H is the height of the equilateral triangle and S is the side length of the hexagon, which can be calculated by using Pythagorean theorem as follows:

$$H = \sqrt{S^2 - \left(\frac{S}{2}\right)^2} = \frac{\sqrt{3}}{2}S. \tag{47}$$

Substituting (47) into (46) yields

$$A_{hexagon} = \frac{3\sqrt{3}}{2}S^2. \tag{48}$$

To evaluate the integral of (10), we split the hexagonal region D, which is shown in Fig. 11, into three regions $D = D_1 \cup D_2 \cup D_3$, then

$$\begin{aligned} E[d_{toCH}^2] &= \int\int_D d^2(x,y)\rho(x,y)\,dx\,dy \\ &= \int\int_{D_1} d^2(x,y)\rho(x,y)\,dx\,dy \\ &\quad + \int\int_{D_2} d^2(x,y)\rho(x,y)\,dx\,dy' \\ &\quad + \int\int_{D_3} d^2(x,y)\rho(x,y)\,dx\,dy. \end{aligned} \tag{49}$$

To find the straight-line equations, which are shown in Fig. 11, we use the point-slope form

$$y - y_1 = m(x - x_1), \tag{50}$$

where m is the slope of the line and is equal to $(y - y_1)/(x - x_1)$, and (x_1, y_1) is one known point on the line. Accordingly, the bounded regions that will define the limits of integration are

$$\begin{aligned} D_1 &= \left\{(x,y): 0 \le x \le \frac{1}{2}S, \frac{\sqrt{3}}{2}S - \sqrt{3}x \le y \le \sqrt{3}x + \frac{\sqrt{3}}{2}S\right\}, \\ D_2 &= \left\{(x,y): \frac{1}{2}S \le x \le \frac{3}{2}S, 0 \le y \le \sqrt{3}S\right\}, \\ D_3 &= \left\{(x,y): \frac{3}{2}S \le x \le 2S, \sqrt{3}x - \frac{3\sqrt{3}}{2}S \le y \le \frac{5\sqrt{3}}{2}S - \sqrt{3}x\right\}. \end{aligned} \tag{51}$$

Therefore, (49) becomes

$$\begin{aligned} E[d_{toCH}^2] &= \rho \int_{x=0}^{x=(1/2)S} \int_{y=(\sqrt{3}/2)S-\sqrt{3}x}^{y=\sqrt{3}x+(\sqrt{3}/2)S} (x-S)^2 + (y-H)^2\,dx\,dy \\ &\quad + \int_{x=S/2}^{x=(3/2)S} \int_{y=0}^{y=\sqrt{3}S} (x-S)^2 + (y-H)^2\,dx\,dy \\ &\quad + \int_{x=(3/2)S}^{x=2S} \int_{y=\sqrt{3}x-(3\sqrt{3}/2)S}^{y=(3\sqrt{3}/2)S-\sqrt{3}x} (x-S)^2 + (y-H)^2\,dx\,dy. \end{aligned} \tag{52}$$

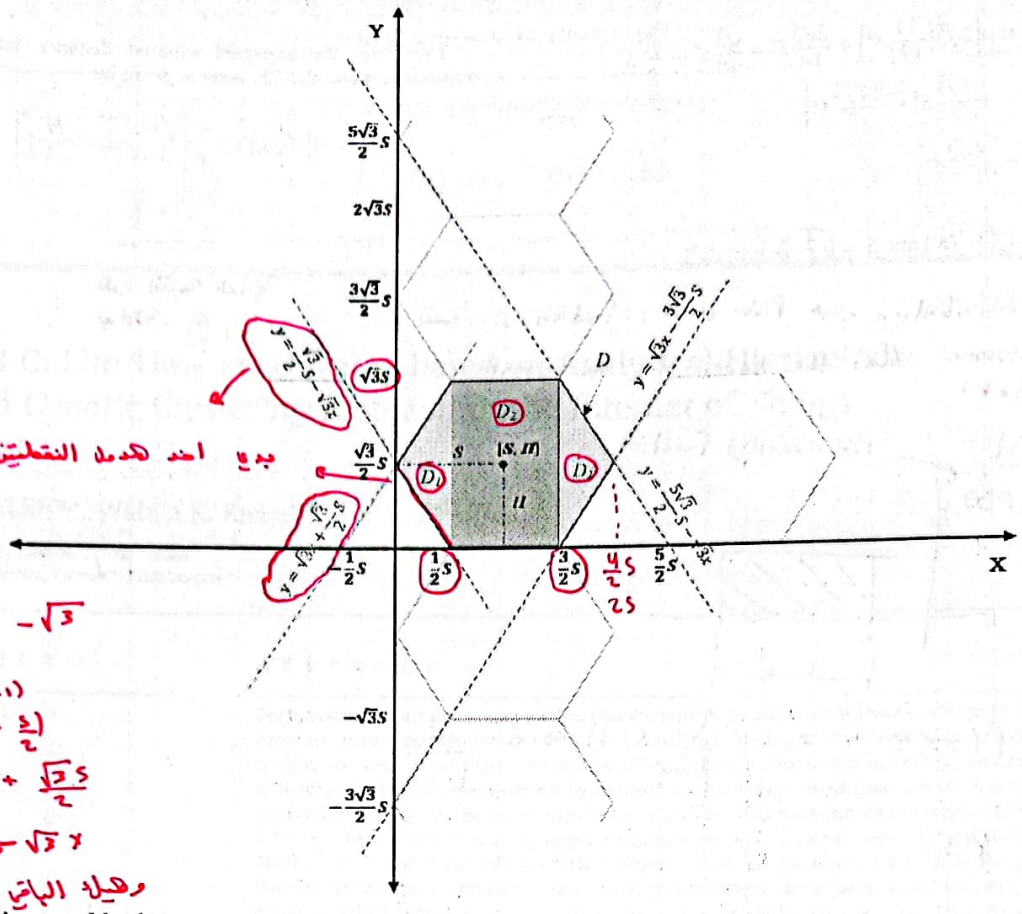


Fig. 11 *Hexagonal region of the cluster*

After integrating with respect to y, the following is obtained:

$$E[d^2_{toCH}] = \rho\left[\int_{x=0}^{x=(1/2)S}(x-S)^2y+\frac{(y-H)^3}{3}\Big|_{y=(\sqrt{3}/2)S-\sqrt{3}x}^{y=\sqrt{3}x+(\sqrt{3}/2)S}dx + \int_{x=(1/2)S}^{x=(3/2)S}(x-S)^2y+\frac{(y-H)^3}{3}\Big|_{y=0}^{y=\sqrt{3}S}dx + \int_{x=(3/2)S}^{x=2S}(x-S)^2y+\frac{(y-H)^3}{3}\Big|_{y=\sqrt{3}x-(3\sqrt{3}/2)S}^{y=(5\sqrt{3}/2)S-\sqrt{3}x}dx\right]. \quad (53)$$

Inserting the upper and the lower limits of y into (53) gives (see (54)).

Upon simplification, (55) is obtained (see (55)). By integrating with respect to x and inserting the limits of integration, (56) can be obtained. Ultimately, the following is provided after simplification

$$E[d^2_{toCH}] = \rho\left[2\sqrt{3}\left(\frac{S^4}{64}-\frac{2S^4}{24}+\frac{S^4}{8}\right)+\frac{2\sqrt{3}S^4}{64} + \left(\frac{\sqrt{3}S(S/2)^3}{3}-\frac{\sqrt{3}S(-S/2)^3}{3}+\frac{\sqrt{3}S^4}{8}\right) + \left(-8\sqrt{3}S^4+\frac{64\sqrt{3}S^4}{3}-20\sqrt{3}S^4+8\sqrt{3}S^4\right) - \left(\frac{81\sqrt{3}S^4}{32}+9\sqrt{3}S^4-\frac{45\sqrt{3}S^4}{4}+6\sqrt{3}S^4-\frac{6\sqrt{3}S^4}{32}\right)\right] \quad (56)$$

$$E[d^2_{toCH}] = \rho\left[\int_{x=0}^{x=(1/2)S}(x-S)^2(2\sqrt{3}x)+\frac{(\sqrt{3}x)^3}{3}-\frac{(-\sqrt{3}x)^3}{3}dx + \int_{x=(1/2)S}^{x=(3/2)S}(x-S)^2(\sqrt{3}S)+\frac{(\sqrt{3}S/2)^3}{3}dx + \int_{x=(3/2)S}^{x=2S}(x-S)^2(4\sqrt{3}S-2\sqrt{3}x)+\frac{2(2\sqrt{3}S-\sqrt{3}x)^3}{3}dx\right]. \quad (54)$$

$$E[d^2_{toCH}] = \rho\left[\int_{x=0}^{x=(1/2)S}2\sqrt{3}x(x^3-2Sx^2+S^2x)+2\sqrt{3}x^3\,dx + \left(\frac{\sqrt{3}S(x-S)^3}{3}+\frac{\sqrt{3}S^3x}{8}\Big|_{x=(1/2)S}^{x=(3/2)S}dx\right) + \int_{x=(3/2)S}^{x=2S}-2\sqrt{3}x^3+8\sqrt{3}x^2S-10\sqrt{3}xS^2+4\sqrt{3}S^3+\frac{2(2\sqrt{3}S-\sqrt{3}x)^3}{3}dx\right]. \quad (55)$$

$$E[d^2_{toCH}] = \rho\left[2\sqrt{3}\left(\frac{11}{192}S^4\right) + \frac{2\sqrt{3}}{64}S^4 + \frac{\sqrt{3}}{12}S^4 + \frac{\sqrt{3}}{8}S^4 + \frac{4\sqrt{3}}{3}S^4 - \frac{19\sqrt{3}}{16}S^4\right] = \rho\left[\frac{\sqrt{3}}{2}S^4\right]. \quad (57)$$

Substituting $\rho = 1/(M^2/K)$ and $S = \sqrt[4]{\frac{4}{27}}\frac{M}{\sqrt{K}}$ into (57).

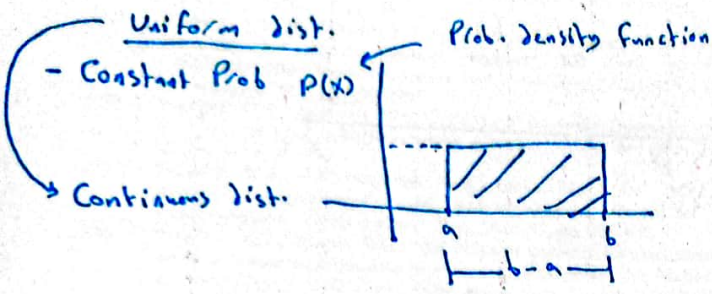
Therefore, the expected squared distance between member sensor nodes and the CH node is given by

$$E[d^2_{toCH}] = \frac{K}{M^2}\left[\frac{\sqrt{3}}{2}\left(\sqrt[4]{\frac{4}{27}} \times \frac{M}{\sqrt{K}}\right)^4\right] = \frac{2\sqrt{3}}{27} \times \frac{M^2}{K}. \quad (58)$$

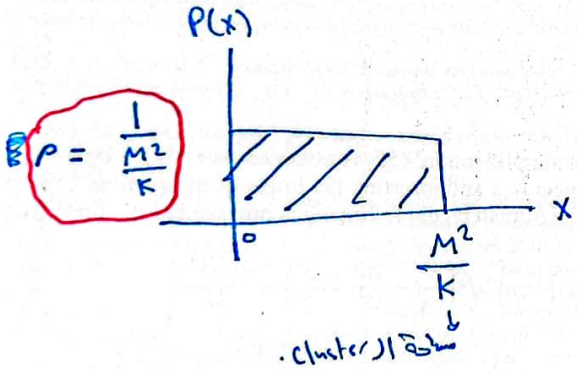
هنا القيمة النهائية بدلالة K, M

ρ = node distribution ⟶ Prob. density function الا نفسها
↳ we assume that the nodes are uniformally distributed.

Uniform dist.
- Constant Prob
→ Continuous dist.



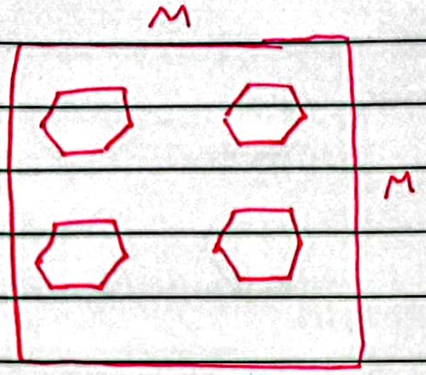
$$P(a \le x \le b) = 1 = \frac{w}{b-a} * H \rightarrow H = \frac{1}{b-a}$$

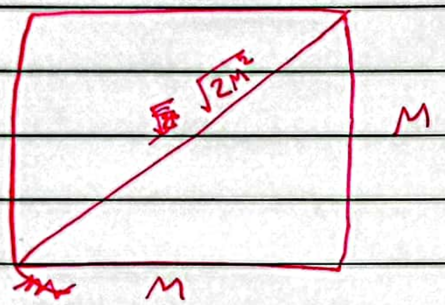


مساحة الـ cluster.

* ممكن يجيب سؤال عالـ Shape ويعمله Shift
واحنا نقسمه ونطلع حدود التكامل بيها

23/11 IoT Network





$$\frac{\sqrt{2M^2}}{2}$$

$$\frac{\sqrt{2}\,M}{\sqrt{2}\,\sqrt{2}} = \frac{M^2}{2}$$

صفحة 3172 اخر اشي

اسئلة بتيجي علينا بالامتحان :

* سؤال صفحة 3180

* تعرف ليه عملنا هاد السيناريو وشو اسبابه.

* بعد هاد التحسين تحسن عندي ال LEACH لانه عندي Kopt و Popt قليل ضمنا انه رح يطلع عندي cluster كل ال node ال CH
وهيك ال energy تاعتهم عالية وقريبة هو الابيكون CH ، وال Popt طلعناها لنضمن وجود cluster بالشبكة.
وعشان مايكون العدد قليل أضفنا للمعادلة : $\frac{E_{CUR}(S_i)}{E_{avg}}$ and $\frac{d_{AVG\_toBS}}{d_{toBS}(S_i)}$

* بجيب سيناريو تبكة فيها cluster وبعطيني معطيات كل واحد وبحكيلي طلعلي مين ال CH.
بكل cluster باستخدام معادلة PT(Si) بس هو يعطيني ال Kopt

* ال Steps لاستخراج ال K-opt : ① نفرض ال Shape للcluster
② بنحسب ال energy المصروفة بال cluster.
③ نحسب ال energy المصروفة بالشبكة.
④ نشتق ونطلع K-opt.

$$E[d^2_{toCH}] = \rho\left[2\sqrt{3}\left(\frac{11}{192}S^4\right) + \frac{2\sqrt{3}}{64}S^4 + \frac{\sqrt{3}}{12}S^4 + \frac{\sqrt{3}}{8}S^4 + \frac{4\sqrt{3}}{3}S^4 - \frac{19\sqrt{3}}{16}S^4\right] \quad (57)$$
$$= \rho\left[\frac{\sqrt{3}}{2}S^4\right].$$

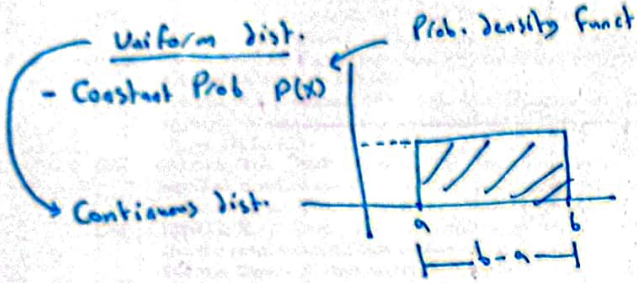
Substituting $\rho = 1/(M^2/K)$ and $S = \sqrt[4]{\frac{4}{27}}\frac{M}{\sqrt{K}}$ into (57).

Therefore, the expected squared distance between member sensor nodes and the CH node is given by

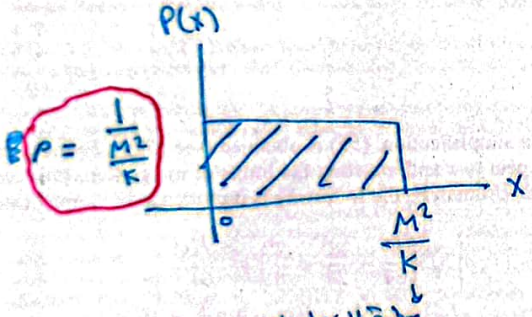
$$E[d^2_{toCH}] = \frac{K}{M^2}\left[\frac{\sqrt{3}}{2}\left(\sqrt[4]{\frac{4}{27}} \times \frac{M}{\sqrt{K}}\right)^4\right] \quad (58)$$
$$= \frac{2\sqrt{3}}{27} \times \frac{M^2}{K}.$$

هذه القيمة الناتجة
بدلالة K, M

ρ = node distribution → Prob. density function أي نفس
As we assume that the nodes are uniformly distributed.



$$P(a \le x \le b) = 1 = \frac{w}{b-a} * H \rightarrow H = \frac{1}{b-a}$$



* ممكن يجيب سؤال عالـ Shape ويعمله Shift
واحنا نقسمه ونطلع حدود التكامل بس

LiM-AHP-G-C: Life Time Maximizing Based on Analytical Hierarchal Process and Genetic Clustering Protocol for the Internet of Things Environment

Khalid A. Darabkh[1], Wafa'a K. Kassab[1], and Ala' F. Khalifeh[2]
[1]The University of Jordan, Amman, 11942, Jordan, k.darabkeh@ju.edu.jo, wafaa.kassab@mof.gov.jo
[2]German Jordanian University, Amman, 11180, Jordan; ala.khalifeh@gju.edu.jo

Abstract— **Contemporary smart sensing paradigms, that are provided via diverse Internet of Things (IoT) mechanisms, propagate across considerable domains of daily life, such as health, agriculture, transportation, trade sectors to urban environments and smart cities. The new era of revolution in information technology will rely on processing and analyzing big data that are gathered by tremendous numbers of intelligent sensors, which are disseminated in the surrounding regions. However, most of these devices suffer from restrictions on power resources and processing capabilities, which in turn will enforce stringent restrictions on the network operations. In view of this, the development of novel and efficient energy algorithms for IoT paradigm is a challenging issue bearing in mind that the performance of this state-of-the-art network paradigm cannot be handled effectively by the existing techniques or solutions that are utilized in wireless sensor networks. To meet the requirements of maximizing the IoT network lifetime, we address in this work the challenge of IoT networks as of embedding energy-constrained devices by proposing a novel protocol, namely, Life Time Maximizing Based on Analytical Hierarchal Process and Genetic Clustering (LiM-AHP-G-C) protocol. In particular, the proposed protocol presents a novel optimal clustering algorithm for un-rechargeable battery-powered IoT devices, an efficient IoT heads selection algorithm, a heuristic method for optimal hop selection, and a model for avoiding intra- and inter-cluster interferences for IoT networks. The simulation results show that our proposed protocol outperforms the other existing works in terms of network lifetime, resource utilization, and scalability metrics.**

Keywords – **IoT environments; smart city; routing; clustering; AHP; cluster head rotation; throughput; GA; broadcast domain; wireless sensor; IoT communication protocols; multi-band antennas; FHSS**

1.Introduction

The Internet of Things (IoT) term was first coined by Kevin Ashton in 1999, to belong later to the fundamental block for the upcoming intelligent world that paves the way to the information revolution era [1] [2]. In specific, the IoT technology builds a bridge between the cyber domain and the things inside the physical world to permit unprecedented ubiquitous surveillance and intelligent control. The major idea behind IoT is that in the close future, most of the things that surround us will be accessible, sensed and connected inside the dynamic, living, global structure of the Internet [3]. Wireless Sensor Network (WSN) plays a considerable role in IoT as of covering a spacious application range indispensable for the IoT. The IoT networks are similar to WSNs, where the nodes in both networks are battery-powered microsystems embedded with transducers to monitor the surrounding [4]. However, the IoT nodes are embedded with a wireless radio to form a wireless network autonomously and communicate with each other. Nevertheless, these nodes have severe resource constraints in terms of battery power, memory size, computational and communication capabilities [2] [5]. These factors should be taken into account as they are deemed as a blueprint for the prosperity of the entire routing process, which requires optimizing the conventional field estimation and data aggregation methods such as multi-hopping and clustering techniques.

1.1 Problem Statement

Commonly, prolonging the network lifetime depends on the effective management of the energy resources [6]. In fact, the energy consumption is one of the most critical issues that must be taken into consideration while designing IoT networks. Most generally adopted energy preservation techniques, in the near past, are data aggregation, duty cycling, and clustering [7] [8] [9]. Data aggregation approach aims to eliminate any replication in the transmitted data in addition to reduce the number of packets that reach the fog node [10] [11]. Duty cycling is an operational method that is utilized to minify the amount of energy consumption caused by sensor nodes, while being in the idle mode, through periodically placing them in the sleep mode, thereby lowering the duty cycling (i.e., having more power-saving) [12] [13]. The fog computing paradigm is proposed to extend cloud computing services from the core to the edge of IoT networks as of providing a highly virtualized platform that supplies many

clusters and determines the nodes belong to each cluster based on their coordinates. Thereafter, it specifies the heads and relaying nodes (if needed) for each cluster.

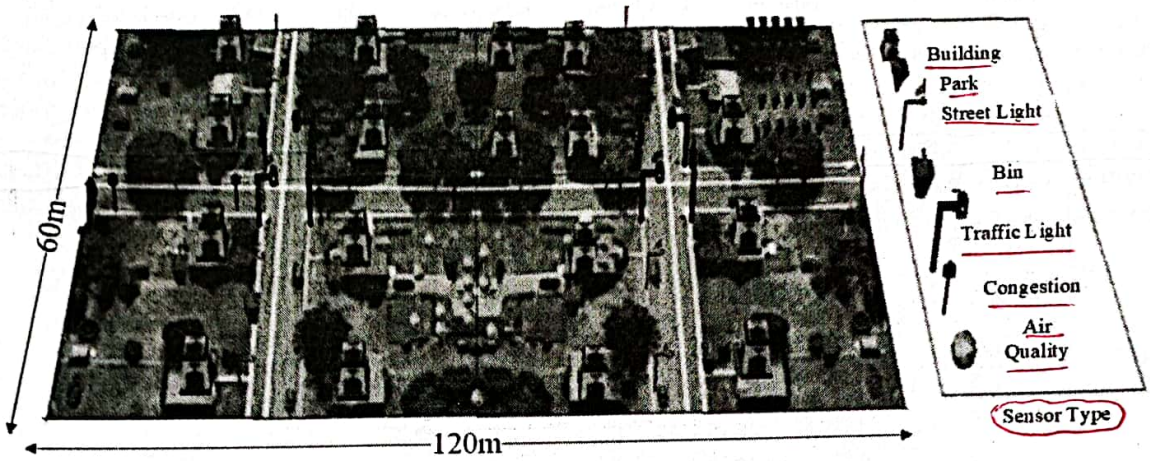


Figure 1. A 60 × 120 IoT area from google earth

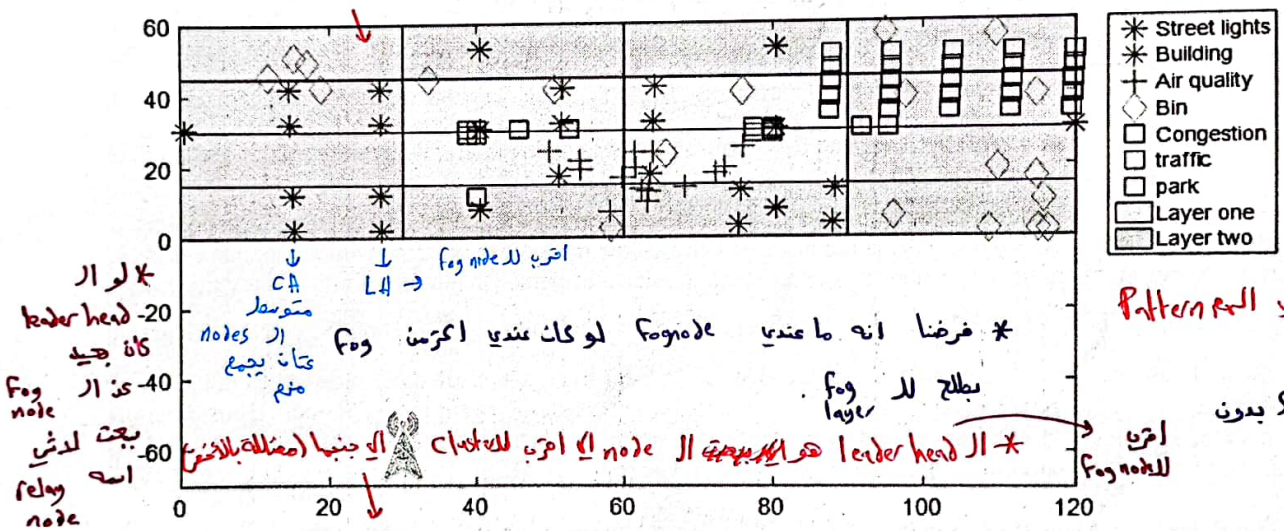


Figure 2. IoT network model in MATLAB

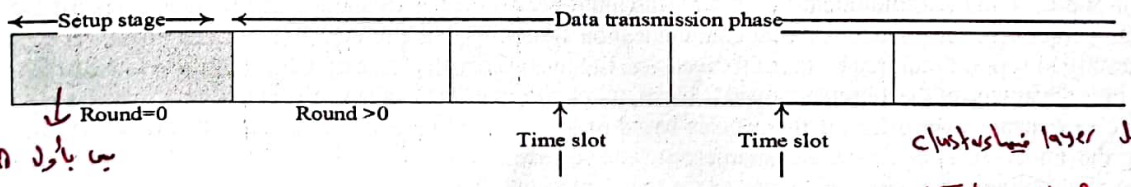


Figure 3. The division of IoT network lifetime

3.3 Division of Network Area

Our major concern is how to connect the width and height of clusters with the threshold distance (i.e., the distance where beyond it, the two-ray channel propagation model is used). In specific, the width and height of clusters should be chosen considering the following constraints:

1. Avoiding the transmissions over the two-ray channel propagation model.
2. Making use of the multi-hop routing algorithm proposed in this work.

It is worth mentioning that avoiding the transmissions over two-ray channel propagation model, keeping the transmissions using Friis free-space channel propagation model, maintains having lower transmit power, thereby prolonging the IoT network lifetime. On the other hand, making the multi-hop routing algorithm proposed in this work feasible, the width of the cluster should be much less than the threshold distance. In light of the above, we want to draw the attention to the point that based on empirical and experimental works, we are eventually able to come up with expressions that describe the division of the network area proposed in this work. Particularly, the height of a cluster is half of its width while the width of a cluster should not exceed 30% of the threshold distance. In addition, we formulate an expression that relates the network length (L_n) with the cluster length (L_c) to have the number of layers (N_l) in the network area as shown below:

$$N_l = ceil(\frac{\rho}{2}), \tag{1}$$

where:

$$\rho = ceil(\frac{L_n}{L_c}). \tag{2}$$

Moreover, the total number of clusters (N_c) in a network area is represented by:

$$N_c = (2N_l)^2. \tag{3}$$

Interestingly, the number of clusters in a layer (N_{c_l}) is expressed as:

$$N_{c_l} = \delta + \sigma \times (L_{num} - 1), \tag{4}$$

where, δ and σ represent the coefficients that have been selected experimentally, while their values are listed in Table 2 (i.e., simulation parameters table). L_{num} refers to the layer number.

It cannot be missed out to be mentioned that the severe resource of energy consumption in IoT nodes is communication, which highly depends on the distance between the transmitter and receiver [35]. In our work, we follow the same energy radio and channel propagation models that are adopted in [36-38]. Similar to its counterparts [20] [28] [29], the LiM-AHP-G-C protocol, considers that a LH consumes $E_{DA}(\frac{nJ}{bit}/signal)$ for data aggregation process and compression of multiple data messages of length q-bit each, which are those received from its cluster members along with its own message, into one single message to be sent afterward to its CH node. Ultimately, the CH node aggregates that compressed message along with its own data message and consequently transmits a compressed version toward the fog node.

3.4 The Broadcast Algorithm

In the evolving telecommunication world, the multi-band antennas technology has become a desired and significant component for numerous recent communication systems because of their eye-catching properties such as their ability to support multiple bands of frequencies, lightweight, small size, easy fabrication, and low cost [39] [40]. To take advantage of the aforementioned features, we employ such type of antennas in this work, which allows each node to transmit over different frequencies based on its type and the required transmission range, thereby avoiding the intra and inter-cluster communication interferences. Attractively, the fog node is responsible for preparing all the important parameters, required per round, before the beginning of the data transmission stage, such as the layers' ID along with their clusters' IDs, members and heads of each cluster, as well as relay node ID of each cluster (if needed). Consequently, it organizes these parameters into a table to be then broadcasted within the area of its dominance, which allows the nodes further to discover how far they are from it relying on the received signal strength indication. Notably, Fig. 4 shows a portion of a control packet belongs to network clusters of layers 1 and 2 from round 494 to 505.

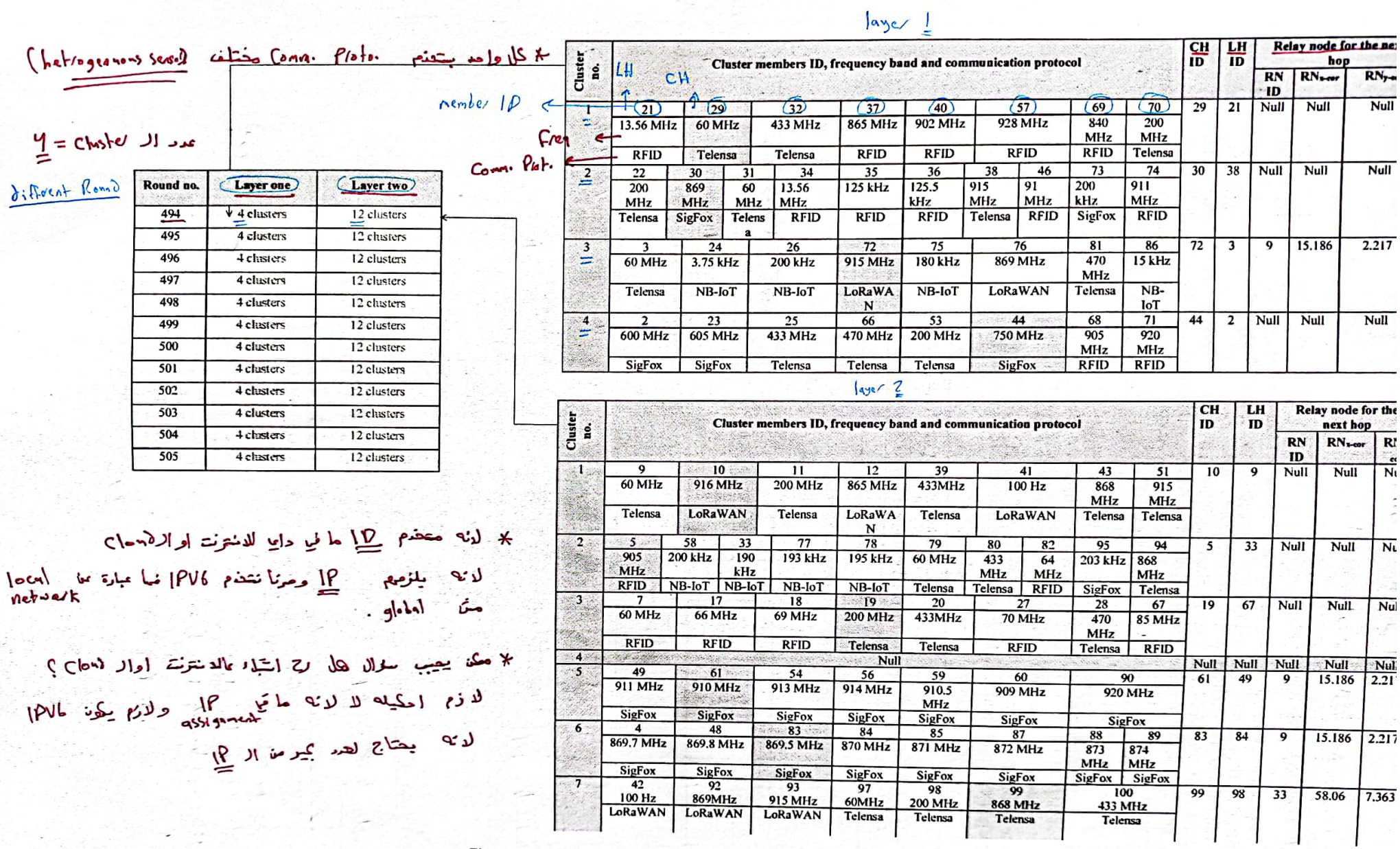


Figure 4. A portion of control packet broadcasted by fog node at the setup stage

3.5 AHP Heads Selection Algorithm

In this paper, a novel cluster head and leader head selection scheme, which is based on the AHP algorithm, is performed by the fog node due to having unrestricted power source along with high processing speed and huge storage capabilities. The AHP is an effective approach in dealing with complex decision-making processes as of being able to help the decision-makers to place priorities and select the optimal option through splitting complex choices into a series of pairwise comparisons and finally synthesizing the results [41]. Moreover, the AHP incorporates a valuable method for checking the consistency of the decision maker's assessments, hence, lessening the bias in the decision-making process [42]. In this work, the AHP is utilized to deal with both CH and LH selection through considering the following steps:

Step 1: Structuring hierarchy

The objective of the decision, which is selecting an optimal CH and LH per cluster, is prioritized at the highest level of the hierarchy as shown in Fig. 5. The next level comprises of the decision variable (dimensions), whereas the least level includes all the IoT sensors that need to be assessed (alternatives).

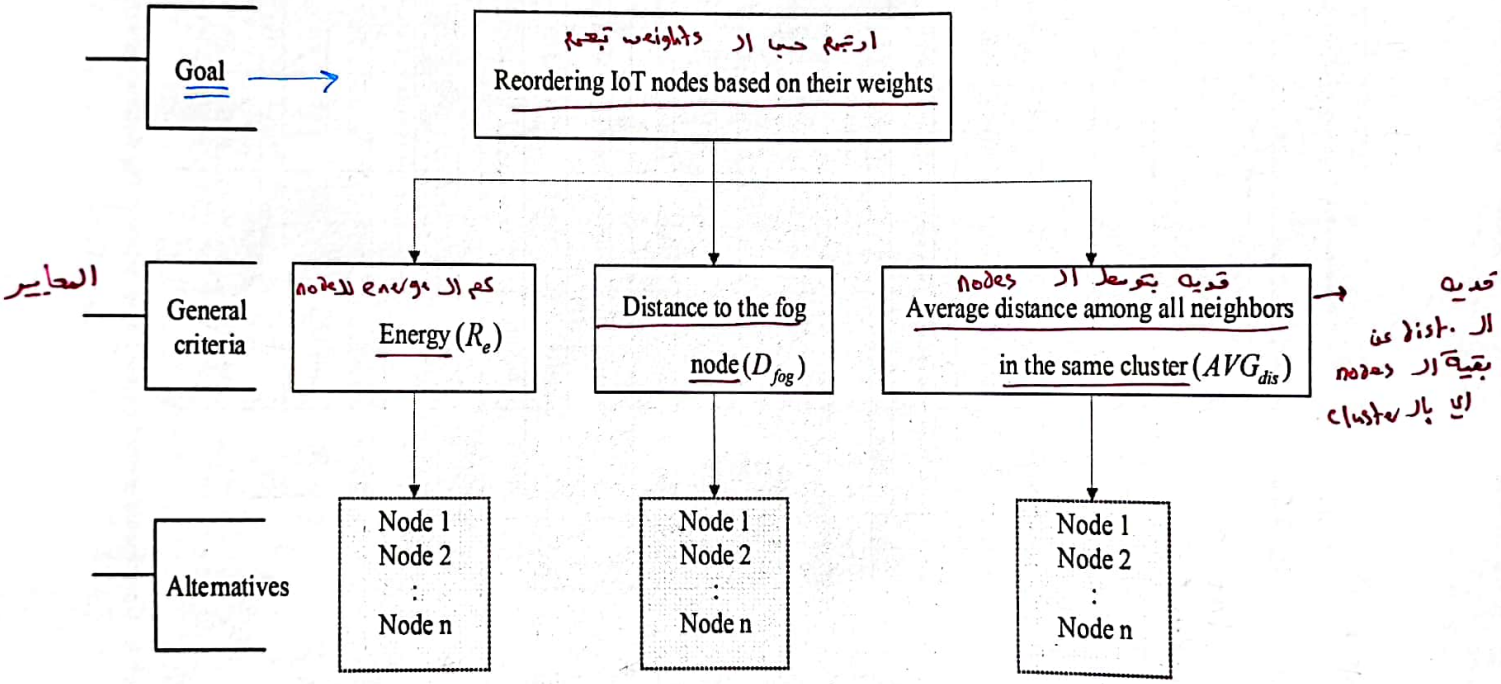


Figure 5. Structuring AHP hierarchy for CH and LH selections

Step 2: Calculating the local-weight vector and consistency check

A local weight refers to the weight of each dimension and accordingly will be as an entry of the local-weight vector. However, the approach of finding the local-weight vector is illustrated below.

a) *Making pairwise comparisons*: In order to compute the weights of distinctive dimensions, the AHP algorithm begins with creating a pairwise comparison matrix A as shown below:

$$A = \begin{pmatrix} a_{R_eR_e} & a_{R_eD_{fog}} & a_{R_eAVG_{dis}} \\ a_{D_{fog}R_e} & a_{D_{fog}D_{fog}} & a_{D_{fog}AVG_{dis}} \\ a_{AVG_{dis}R_e} & a_{AVG_{dis}D_{fog}} & a_{AVG_{dis}AVG_{dis}} \end{pmatrix}, \quad (5)$$

Taking into consideration that A is $m \times m$ matrix, where m denotes for the number of the evaluated dimensions which include the residual energy of an IoT node (R_e), the distance between that node and the fog node (D_{fog}), and the average distance between that node and other cluster members (AVG_{dis}). Each entry in the matrix A (i.e., a_{ij}) represents the importance of the i^{th} dimension relative to the j^{th} dimension. For instance, if $a_{ij} > 1$, then the i^{th} dimension will be more important and favorable than the j^{th} dimension. Similarly, if $a_{ij} < 1$, then i^{th} dimension will be less important than the j^{th} dimension, while $a_{ij} = 1$ indicates that both i^{th} and j^{th} dimensions have the same importance. Based on the above, the relative importance among the two dimensions (i, j) is determined according to a numerical scale that goes from 1 to 9, as shown in Table A.1 at appendix A. On the other hand, the exact value of the pairwise comparison matrix that is adopted in our simulations is provided in (A.1), found in appendix A.

b) *Calculating the local weight vector:* Generally, the local weight vector considering all dimensions can be found as

$$W = \begin{pmatrix} w_{R_e} \\ w_{D_{fog}} \\ w_{AVG_{dis}} \end{pmatrix}, \tag{6}$$

Specifically, the local weight vector of dimension i (w_i) is computed by finding the average of every row in the normalized matrix A_{norm} (from matrix A), which is as

$$w_i = \frac{\sum_{k=1}^{m} \bar{a}_{ik}}{m}, \tag{7}$$

where $\bar{a}_{ij}$ is the normalized a_{ij} and can be computed as

$$\bar{a}_{ij} = \frac{a_{ij}}{\sum_{k=1}^{m} a_{kj}}. \tag{8}$$

It is good to mention that the summation of each column in A_{norm} should be equal to 1.

c) *Checking for consistency:* A matrix A is considered to be consistent as long as $a_{ij} \cdot a_{ji} = 1$ Hence, to check the consistency of any matrix, we should find the Consistency Ratio (CR), which is expressed as

$$CR = \frac{CI}{RI}, \tag{9}$$

where, CI represents the consistency index, which is expressed as shown in (10), while RI refers to the random index which can be selected from Table A.2, shown in appendix A, based on the number of dimensions chosen.

$$CI = \frac{\lambda - m}{m - 1}, \tag{10}$$

where, λ refers to the eigenvalue of the pairwise comparison matrix and is computed as

$$\lambda = \frac{\sum_{k=1}^{m} CO_k}{m}, \tag{11}$$

where, CO represents the consistency vector and can be computed as

Appelix A مثال

$$CO = w_s \times \frac{1}{w}, \tag{12}$$

where, w_s denotes for the weight sums vector and can be calculated as

$$w_s = A \times w. \tag{13}$$

To this end, we can determine whether the comparison matrix is consistent or not based on the following constraint. If $0 \le CR < 0.1$, then the local weights will be consistent, otherwise, they will be not consistent. If this constraint is not met, then a recalculation is required. In other words, the levels of importance among the three dimensions should be reconsidered to eventually meet this consistency check constraint.

Step 3: Calculating the global-weight vector

The global-weight vector is calculated for each cluster by finding the aforementioned three dimensions for each node in a cluster and then including them in a matrix where the number of its rows refers to the total number of cluster members. Thereafter, a normalized version of this matrix is considered to be then multiplied by the local-weight vector, found in step 2, which ultimately results in obtaining the global weight vector. Interestingly, in our proposed protocol, the control message, sent by the fog node, has a massive information in which the nodes of each cluster are ordered in a list in a descending order based on their global weights. Actually, the top two nodes will serve as a CH and LH, respectively, for that cluster in the current batch (i.e., group of rounds) while the next top two nodes will serve as a CH and LH, respectively, for that cluster in the next batch and so forth. The batch duration is determined by the fog node which is assumed to be in a knowledge of all messages' exchanges between all nodes in the network along with their costs. In addition to aforementioned control message information, the round numbers at which the nodes alternate their heads are included. The criterion of alternating among batches is as follows. Both heads selected will stay serving in their roles as long as their energy consumption, until the current round, equals or does not exceed an energy threshold, E_{Thrs}. This criterion helps in maintaining a balance in the energy consumption among cluster's nodes. Literally, the global-weight list is subject to change as soon as any of the heads ran out of energy. In other words, the informative control message, sent by the fog node, includes further the new global-weight lists that should be considered by all clusters' members along their activation rounds. The main reason of making the whole task to be handled by the fog node is to reduce the control overhead which will be required, from sensor nodes, for achieving this purpose.

3.6 Genetic Relay Node Selection Algorithm

The genetic algorithm is an adaptive heuristic search algorithm that mimics the genetic concepts of natural selection, mating, mutation, and inheritance and is widely utilized in optimizing plenty of research problems [43]. The genetic algorithm starts with a set of randomly generated probable solutions, known as the initial population. Each individual is represented through an array or a string of genes called a chromosome, where the length of all chromosomes in the population should be equal. Every permissible gene should be evaluated via a fitness function to estimate its efficiency. Hence, the fitness function has to be formulated in such a way that a permissible gene provides a result around the optimum solution.

In the context of IoT sensors and for any cluster, if the distance between the CH and fog node exceeds the threshold distance, then the employment of relay nodes will become feasible, as shown in Fig. 6. Therefore, all LHs or CLs in the network are considered initially as candidate relay nodes. Upon satisfying a criterion, those filtered heads will turn out to be permissible relay nodes. This criterion involves three conditions where the first condition relates to that, for any cluster the distance from the corresponding CH to a candidate relay node should not transcend the threshold distance. This can be justified due to making sure that there is no need to consider any further relay in between. The second condition refers to that the distance from a candidate relay node to the fog node should be less than the distance from the CH to the fog node. This condition is chosen to insure that the candidate relay node should be in the direction toward the fog node. The last one relates to that the energy of the candidate relay node should be greater than or equal to the average energy of all candidate relay nodes. As mentioned when discussing the AHP heads selection model, the fog node is in charge of finding the relay node for every CH or CL in any round, as shown in Fig.4. In other words, the control message, sent by the fog node, will keep all CHs or CLs informed of their optimal relay nodes in every round. That is why the energy concern is included in the aforementioned criterion.

However, for any cluster, there might be a number of possible hops for forwarding its traffic toward the fog node (RE_{num}), which can be expressed as

$$RE_{num} = ceil\left(\frac{\left(\left(y_{fog} - y_{CH}\right)^2 + \left(x_{fog} - x_{CH}\right)^2\right)^{0.5}}{d_{Thres}}\right). \tag{14}$$

After finding the permissible relay nodes for all clusters in the network, the initial population can be determined, which consists mainly of many chromosomes. Indeed, each chromosome is made up of many genes where each gene belongs to one cluster and, in this context, refers to the permissible relay nodes. For instance, imagine there is a network in which four clusters are formed. This implies having 4 genes where each gene has a number of permissible relay nodes. As a follow-up to our prior example, the 4 genes, in sequence, have 2, 4, 3, and 5 permissible relay nodes, respectively. At this case, the initial population includes 5 chromosomes where the 5 permissible values, belong to the last gene, are spread out evenly among those chromosomes. In regards to the other three clusters, the permissible values are distributed alternatively among the 5 chromosomes. At this stage, finding the fitness value of each gene in each chromosome is initiated. The fitness function adopted in our algorithm (FN) is expressed as:

$$FN = \alpha \times f_1 + \beta \times f_2 + \gamma \times f_3, \tag{15}$$

where f_1 represents the ratio of the energy of the current permissible relay node ($E_{res}(pe)$) to the average energy of all permissible relay nodes ($E_{avg-res}$). This reflects that the possibility of the current permissible relay node to be the optimal relay node gets high as its energy level is high. Moreover, f_2 denotes for the ratio of the average distances from all permissible relay nodes to the fog node, to the distance between the current permissible relay node and the fog node. As it gets closer to the fog node, the possibility of the current permissible relay node to be the optimal relay node increases. In addition, f_3 refers to the ratio of the average number of times that all permissible relay nodes have become optimal relay nodes to the number of times the current permissible relay node played this role. In fact, this is an important concern. In different words, to achieve an efficient load balancing, if the current permissible relay node is already served in the prior rounds as the optimal relay node, then its possibility to serve longer will decrease. Lastly, α, β and γ are the fitness function coefficients which are experimentally chosen and listed in Table 2.

In light of the above discussion, f_1 is expressed by

$$f_1 = \frac{E_{res}(pe)}{E_{avg-res}}, \tag{16}$$

Furthermore, f_2 is given by

$$f_2 = \frac{\dfrac{\sum_{i=1}^{N}((y_{fog} - y_i)^2 + (x_{fog} - x_i)^2)^{0.5}}{N}}{((y_{fog} - y_{pe})^2 + (x_{fog} - x_{pe})^2)^{0.5}}. \tag{17}$$

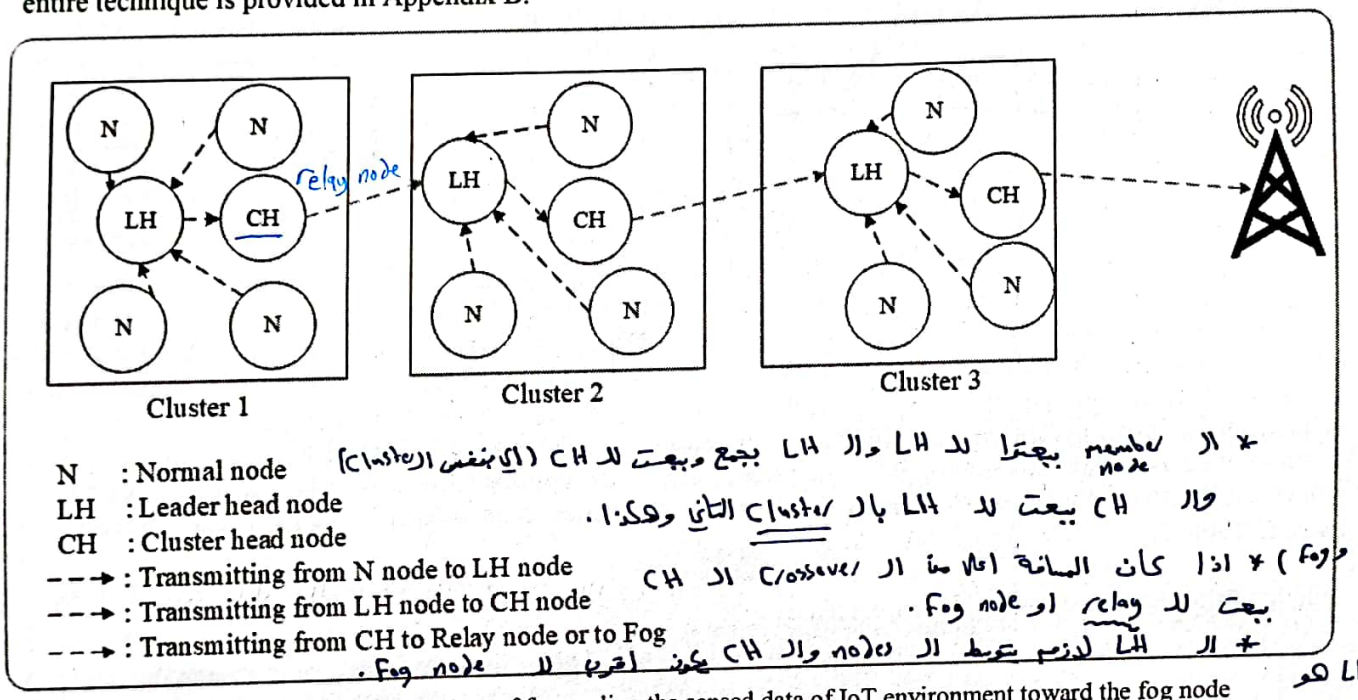
At the end, f_3 is expressed by

$$f_3 = \frac{\dfrac{\sum_{i=1}^{N} C(i)}{N}}{C(pe)}, \tag{18}$$

where, $C(i)$ refers to the number of times that a permissible relay node i has become optimal relay node. N refers to the population size (i.e., the number of permissible relay nodes).

كل جين في الـ Fitness Function بطلعها الـ

At this point, the average of the fitness values of each chromosome is determined. If the average is beyond the boundaries (limits), then this chromosome will be excluded from the initial population. Genuinely, these limits are set based on the minimum and maximum acceptable values of the aforementioned fitness function. As soon as this task gets performed, the process of crossover is started taking into account that there are a lot of strategies available in the literature for this purpose out of which 1-point, k-point, shuffle, reduced surrogates, blend, uniform, heuristic uniform, and discrete crossover where the latter is employed in this work [43], which results in having new children. At this instance, those children will be the new population. After conducting a crossover process, there is a possible mutation outcome where there are many mutations approaches currently available such as bit flip, swap, scramble, and random resetting mutation, which is adopted in this research [44]. Attractively, after a mutation process, there might be a new gene which did not exist among the parents' genes, which is quite realistic. The process of conducting crossover operations and then mutations will be repeated in all generated children till to get one child (chromosome) which represents the optimal solution, where the whole procedure is demonstrated in Fig.7. In other words, it represents the optimal relay node for every CH or CL. A complete example to show the entire technique is provided in Appendix B.



* الـ member node بعتبرا للـ LH والـ LH بجمع وبعتت للـ CH (اي بنفس الـ cluster)
والـ CH بعتت للـ LH بالـ cluster التاني وهكذا.
* اذا كانت المسافة اعلا من الـ crossover الـ CH (بين CH و Fog) بعتت للـ relay او Fog node.
* الـ LH لازم يتوسط الـ node والـ CH يكون اقرب للـ Fog node.
* الـ LH هو بعين Relay من الـ cluster التاني.

Figure 6. The procedure of forwarding the sensed data of IoT environment toward the fog node

3.7 Avoiding Intra- and Inter-Cluster Interferences Algorithm

The rapid proliferation of IoT technology leads to the emergence of a massive number of wireless communication technologies, such as RFID, Telensa, SigFox, LoRaWAN and NB-IoT that could share the same frequency bands in numerous scenarios. Based on this, many signals will be carried through the same frequency band which may lead to the data loss, higher delays, intermittent network connections, and lower network throughput due to signals' interferences that are caused by intra- and inter-cluster communication [45]. To eliminate these interferences, we utilize sensors which are embedded with multi-band antennas in both transmitting and receiving circuits, thereby enabling the use of different communication protocols. These protocols support the Frequency-hopping Spread Spectrum (FHSS) technique in order to enable the fog node to oblige all the sensors in the network area to utilize different frequencies and hence evade any possibility of communication jamming. To achieve these objectives, we employ the mechanism which is explained below and shown in Fig. 8.

a) The inter-cluster interferences, that could happen at the end of each round, is evaded by compelling each CH or CL to transmit over different frequencies considering the following cases:

* كيف بنقدر نتعامل مع Interference الـ inter والـ intra
* بما انه عندي multiband يعني عندي multipath يعني heterogeneous ← من صفات الـ IoT
* داخل نفس الـ cluster ← intra
* بين الـ clusters ← inter والـ يتأثر هو الـ CH.

- In the case of having homogeneous or heterogeneous CHs or LHs operating over the same or various communication protocols with different frequencies, then there will be no possibility for any interference.
- If the CHs support different or identical communication protocols that operate over the same range of frequencies, then the fog node will apply FHSS technique to let these heads transmit over distinctive frequencies.

b) The signals' interferences, which may be caused by the intra-cluster communications among the members of a cluster, is possible to be avoided. In particular, after assigning specific frequencies to all CHs in the network area, as mentioned in the first step, the fog node will assign frequencies to cluster members bearing in mind the following cases:
   - If the members of a cluster have different or the same communication protocols and operate over the same range of frequencies, then the fog node will force them to transmit over distinctive frequencies excluding the frequency that is reserved for their CH.
   - If the cluster's sensors operate over distinctive or similar communication protocols that support different frequencies, then there will be no interference.

c) Due to obeying a batch-based style, the fog node switches the CH role among CHs in the network irregularly. In other words, not all the CHs will serve the same number of rounds (i.e., the batch length is different). Therefore, in our algorithm, the fog node, when preparing the setup parameters, is aware of all frequencies assigned to all nodes in the network in every round, hereby avoiding any interference that may arise by intra- or inter-cluster communications. However, Fig. 9 summarizes all algorithms considered in our proposed protocol.

4. Simulation Results and Discussion

In this section, a massive number of simulations has been conducted for evaluating and analyzing the performance of our proposed protocol. In other words, we firstly introduce the simulation environment and parameters used. Secondly, we define the performance metrics that are used in showing to what extent the contributions of our proposed protocol are significant. Finally, the simulation results along with necessary decisions are provided.

4.1 Simulation Environment

The interest in enhancing different factors and parameters that affect the performance of IoT environments by the researcher's community is rapidly increasing. Accordingly, numerous experimental and simulation platforms have been developed due to being almost impossible to proceed in a real environment where many burdens, that face the setup and implementation in such environments like complexity, time, and cost, may exist. We were eager to choose an environment that can be equipped with different smart services for the reason of considering different scenarios of smart city applications. Consequently, we choose an informative geographical territory, which is Birdsboro borough, where different smart sensing technologies can be deployed and managed easily, hence, improving the life quality of citizens and tourists. Actually, different IoT applications may be considered in this simulated region such as smart traffic management, smart structural health of buildings, smart air quality management, smart parking, smart street lighting, etc. In this research, we use four focal software applications, where various types of sensors are deployed based on specific distributions that refer to the sensor type and its transmission range, which include Google Earth Pro v7.3.2.5776, OpenStreetMap web application, Autodesk AutoCAD vM.107.0, as well as MATLAB R2015a v8.5.0.197613. In our simulation experiments, 100 IoT nodes are deployed using various distributions, as summarized in Table 1, over an area of the size 60 m × 120 m. Furthermore, the simulation parameters used are provided in Table 2 and mainly considered as the default values unless otherwise mentioned in the discussion. The procedure used to acquire the essential information to conduct the simulations, such as buildings, parking lots and street coordinates is illustrated as:

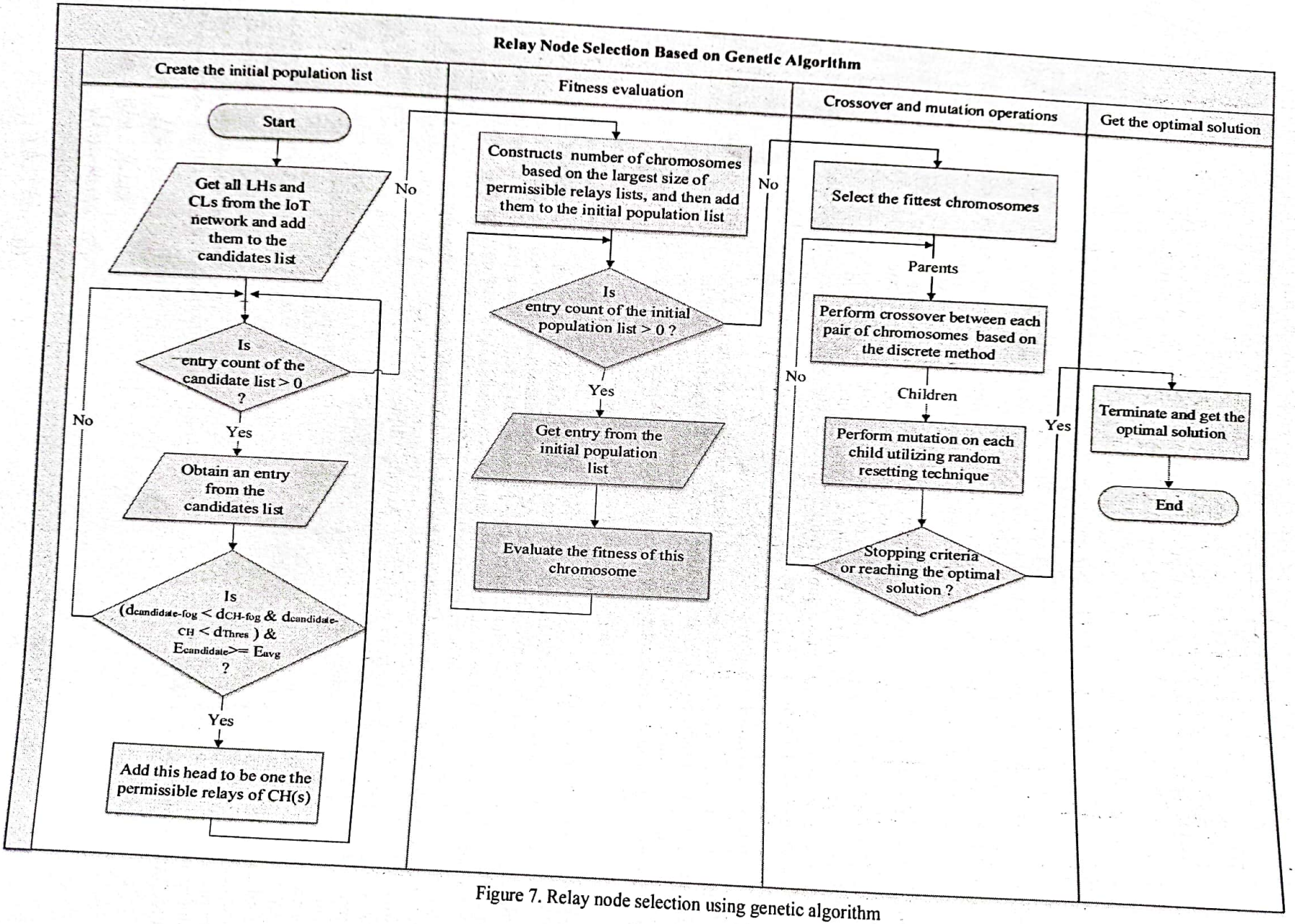


Figure 7. Relay node selection using genetic algorithm

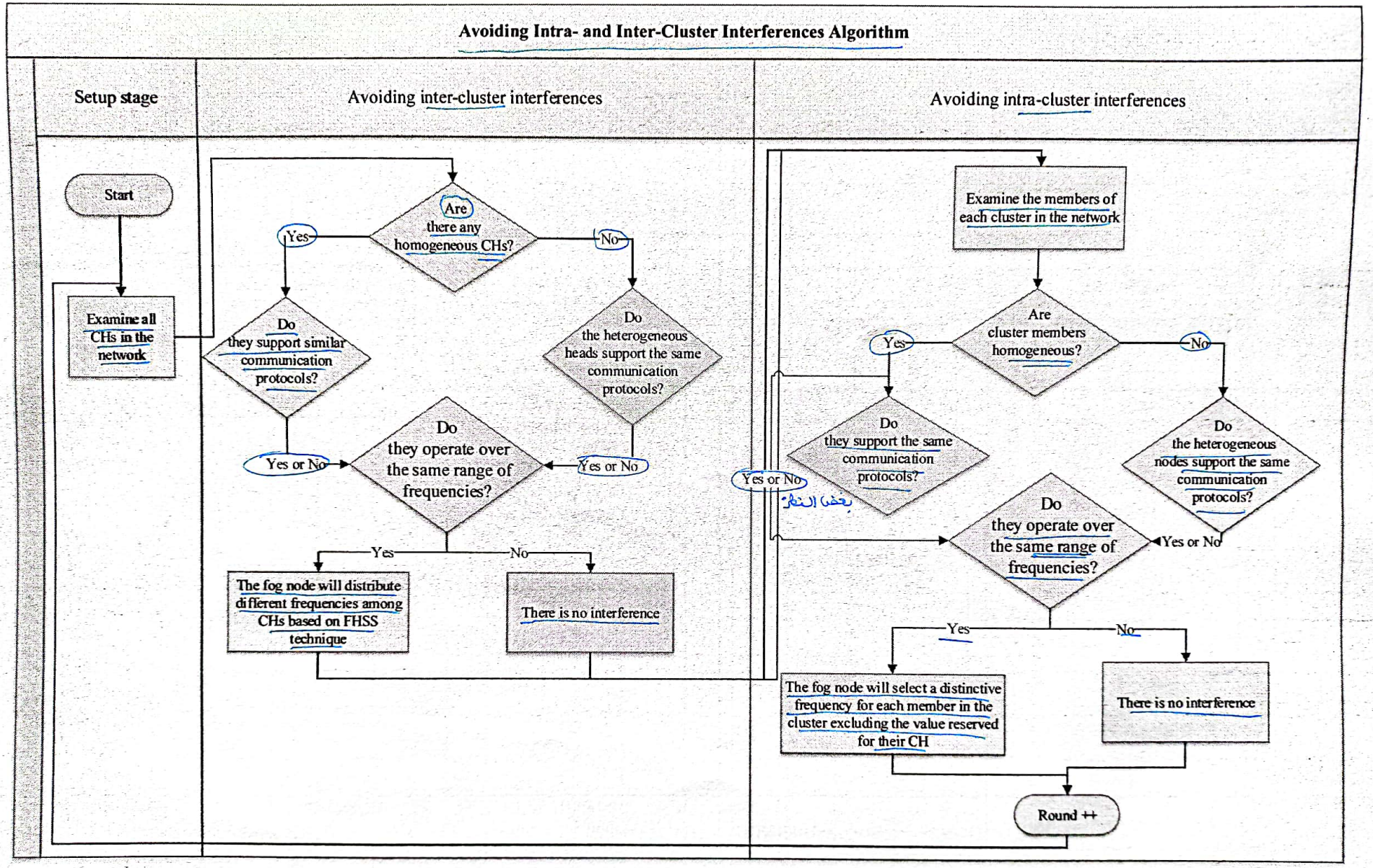


Figure 8. Evading the interferences of intra- and inter-cluster communications in IoT network

a) The designated area is specified and selected via Google Earth Pro App, where Birdsboro borough is chosen to be the region of interest, as shown in Fig. 10.
b) An OSM file is exported to identify the coordinates of all streets and buildings of Birdsboro borough using OpenStreetMap, as depicted in Fig. 11.
c) The OSM file is then imported into AutoCAD App after being converted into KMZ format.
d) The coordinates of traffic lights, structural health building, parking sensors, and all streets, as shown in Fig. 12, which are disseminated based on specific distributions, are then exported from AutoCAD into XLS files.
e) After importing all XLS files, generated from AutoCAD, into MATLAB, the coordinates of all aforementioned sensors are now known and accordingly, the simulation process begins as soon as the step below is considered.
f) The locations of the other sensors, which are of interest too and include the street lights, traffic congestion, air quality, and waste management, are determined directly through MATLAB based on their transmission ranges and appropriate distributions, where all considered sensors are detailed in Table 1.

Table 1. IoT sensors and distributions in various network sizes

	IoT sensors	Distribution	Communication Protocol(s)	Number of sensors for (60 m × 120 m)	Number of sensors for (80m × 160m)	Number of sensors for (100 m × 200 m)	Number of sensors for (160 m × 320 m)
1	Smart parking sensors	Uniform distribution	RFID	25	45	70	190
2	Street light sensors	Uniform distribution [46]	Telensa	8	11	14	22
3	Smart traffic light	Roads Intersections	NB-IoT	8	8	8	8
4	Traffic congestion sensors	Random deployment [47]	Telensa LoRaWAN	6	18	31	80
5	Air quality sensors	Gaussian distribution [48]	Telensa	15	27	44	120
6	Waste management sensors (smart bins)	Random distribution	Telensa	20	36	58	155
7	Structural building	Particle swarm optimization algorithm [49]	SigFox	18	33	53	136
Total number of sensors				100	178	278	711

Table 2. Simulation parameters

Parameter	Value
Network size	60 m × 120 m
Cluster dimensions (height and width)	15 m × 30 m
Equation 4 coefficients	δ=4, σ=8
E_{Tlrs}	1mJ
Fitness function coeffitients	α = 0.5, β= 0.4, γ= 0.1
Fog node location	X=30, Y= -60
Number of IoT nodes	100 nodes
Data message size	6400 bits
Control message Size	200 bits

The initial energy of a node	0.5J				
$\varepsilon_{free-space-amp}$	10 PJ / bit / m^2				
$\varepsilon_{two-ray-amp}$	0.0013 pJ/bit/ m^4				
E_{elec}	50 nJ/bit				
E_{DA}	5 nJ/bit				
$d_{crossover}$	100 m				
Communication protocol name	**RFID** [50] [51]	**Telensa** [52]	**NB-IoT** [53]	**LoRaWAN** [54]	**SigFox** [53]
Communication protocol frequencies	(125–134) kHz 13.56 MHz (60-865) MHz (902-928) MHz	60MHz 200MHz 433MHz 470MHz 868Mhz 915MHz	850-900 Hz 3.75 kHz 15 kHz 180-200 kHz	100Hz 869MHz 915 MHz	200 kHz (868 – 869) MHz (902 -928) MHz
Communication protocol transmission ranges	(1-10) cm (1 -30) m	3km 20km	1 km 10 km	(2-5) km 15km	(3–10) km (30–50) km

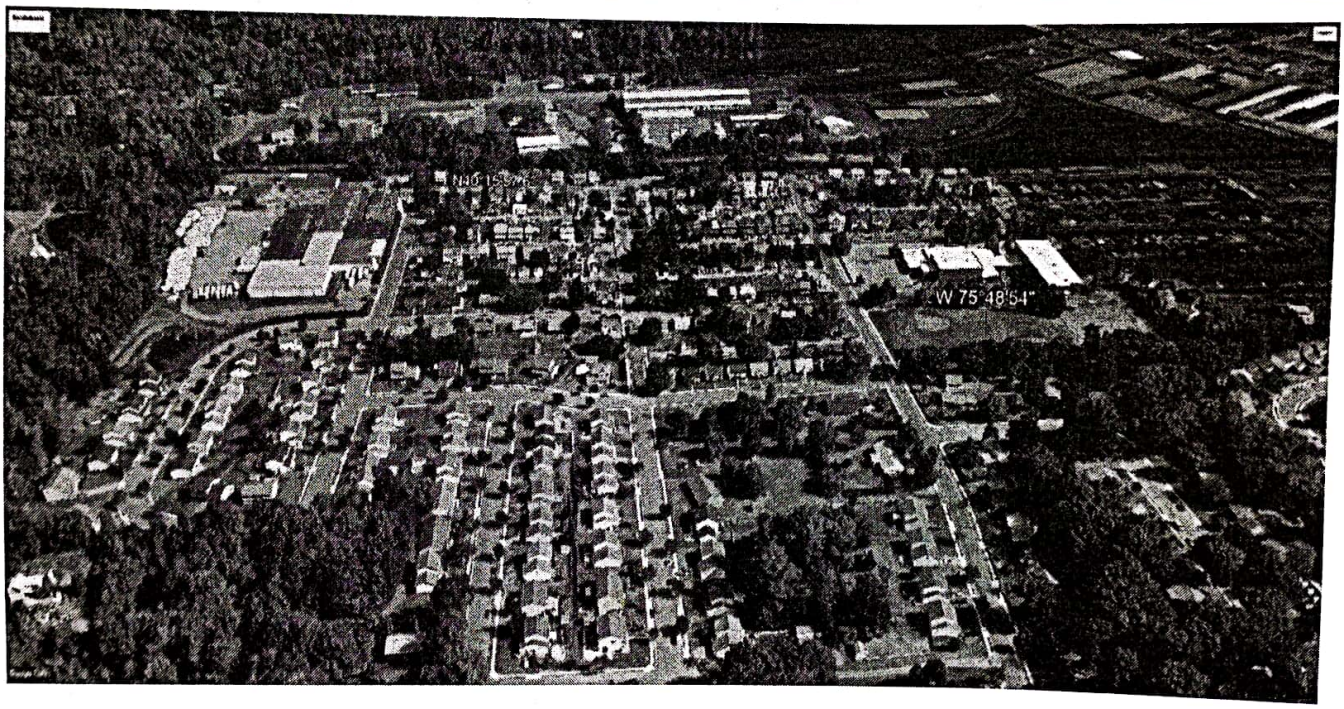


Figure 10. Google earth map of Birdsboro borough in Pennsylvania, United States

4.2 Performance Metrics

The efficiency and robustness of our proposed protocol have been examined thoroughly where a tremendous number of simulations were conducted considering the following performance metrics:

i. Network lifetime which has been assessed in three different perspectives, namely, the First-Node-to-Die (FND), Half-Node-to-Die (HND), as well as Last-Node-to-Die (LND) metrics. The FND denotes for the time gauged (in rounds) from the initial IoT network deployment until the first IoT sensor depletes its energy and subsequently dies. The LND represents the total time (in rounds) till all sensors exhaust completely their energy. Similarly, the HND refers to the number of rounds where exactly half of the sensors deplete their energy.

ii. Network utilization which primarily refers to the ratio of the energy consumed for data transmissions to the energy consumed for both data transmissions and control overhead.

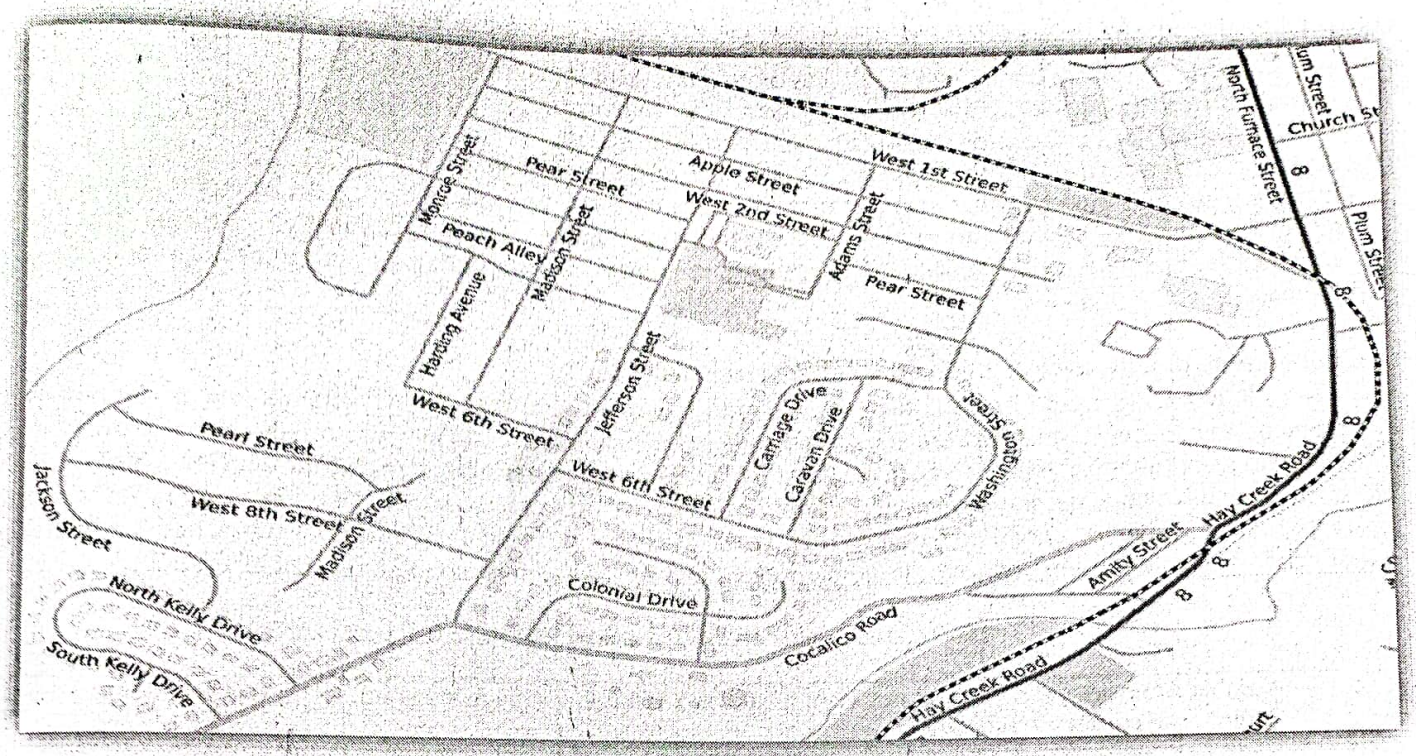


Figure 11. Birdsboro borough buildings and streets by OpenStreetMap

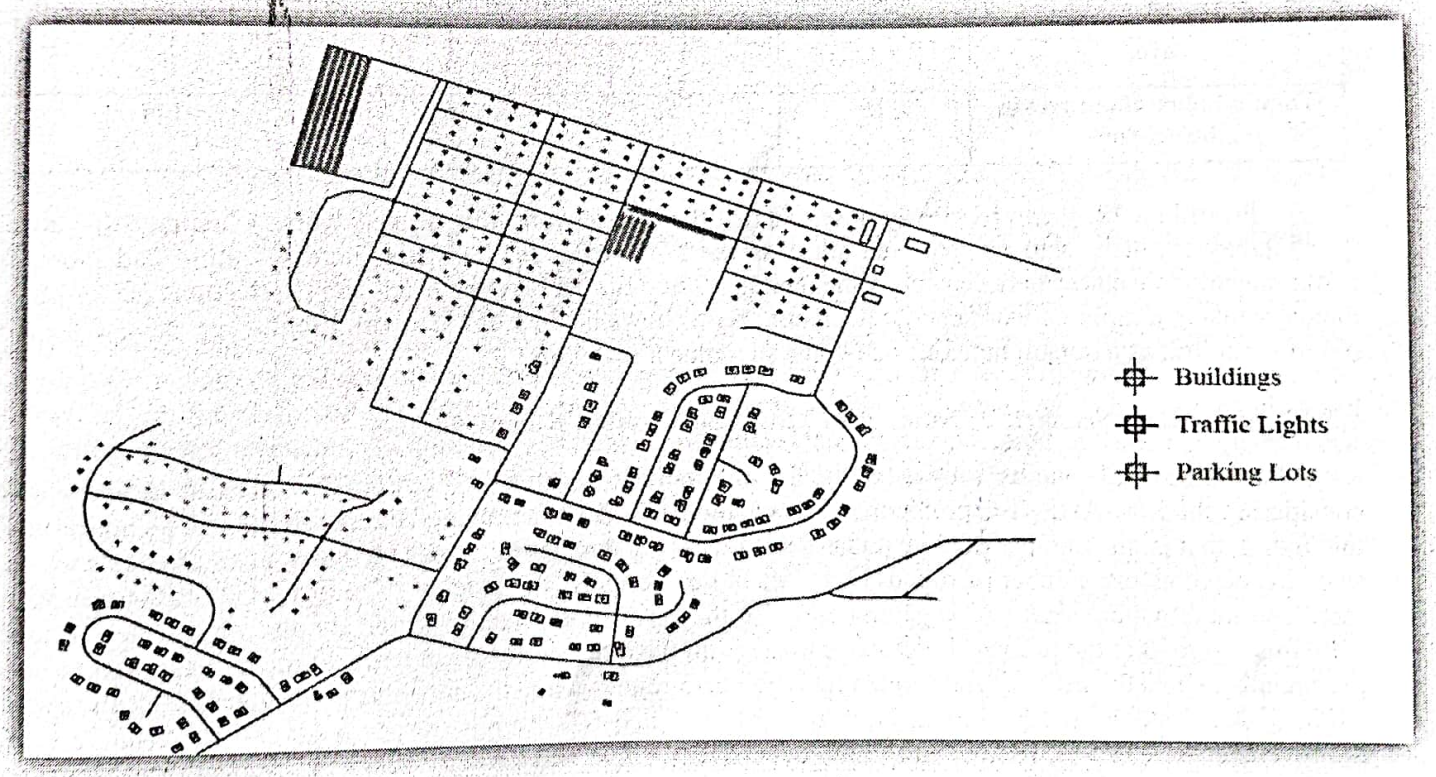


Figure 12. Distribution of different types of IoT sensors through AutoCAD

4.3 Results and Discussion

To validate the scope of correctness of our proposed protocol, many scenarios are considered. Firstly, we investigate the effect of enlarging the network size on the IoT network lifetime of LiM-AHP-G-C protocol and consequently declare the network size that maximizes the IoT network lifetime to be then considered in the

cluster members based on a batch-based time-division approach lead to distributing the energy consumption evenly among sensor nodes. Finally, the usage of genetic algorithm for selecting relay nodes to forward the aggregated data from CHs, which are distant more than the threshold distance away from the fog node, maintains the undergo of Friis free space channel propagation model.

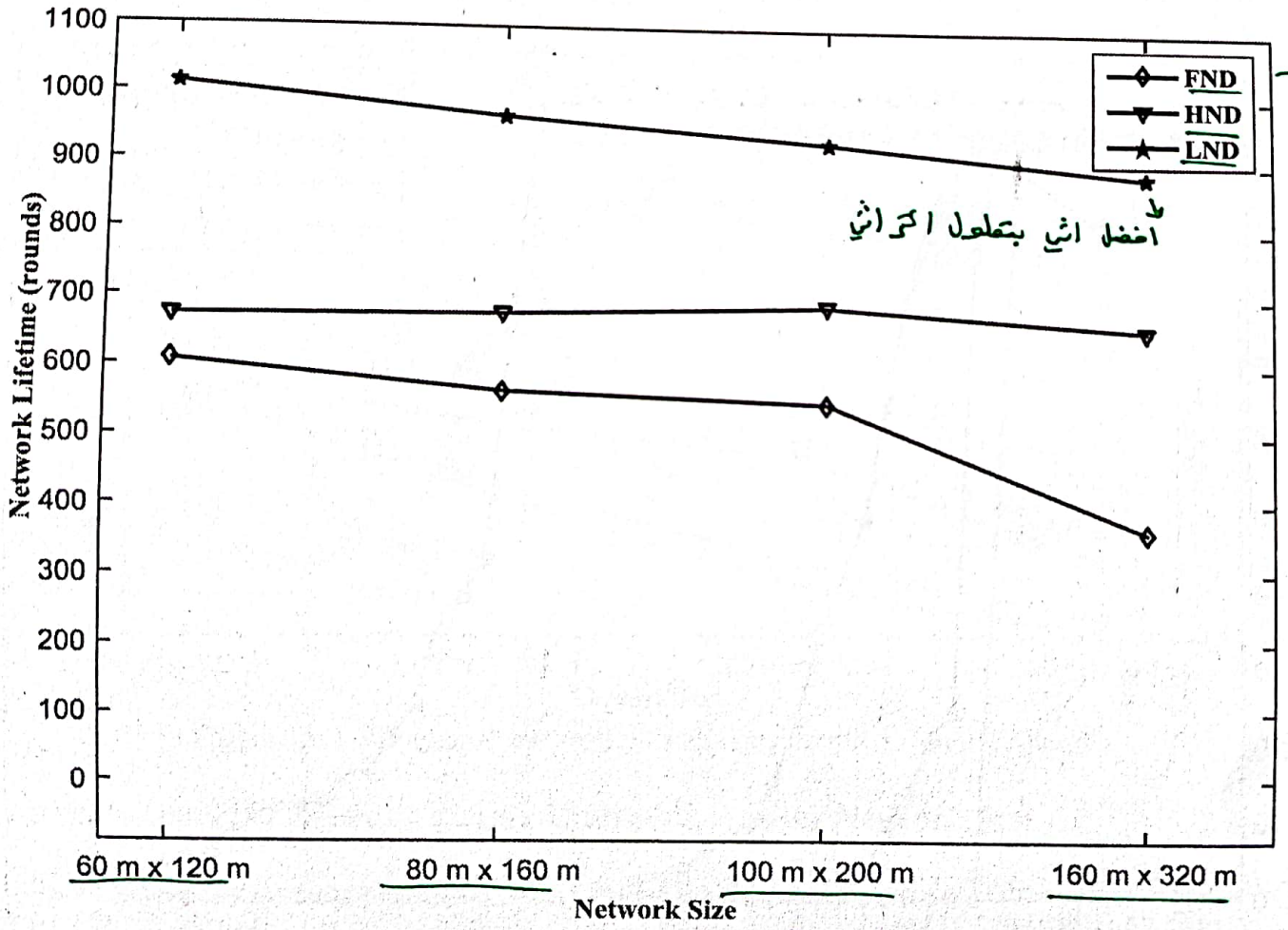


Figure 13. Network lifetime of LiM-AHP-G-C protocol versus different IoT network sizes

The EA-CRP performs better than COCA and UCR protocols which is attributed to employing a multilayered structure, where the sizes of layers decrease toward the fog node. As a consequence, the EA-CRP

letting the nodes to exhausted their energy so early. It is good to stress on the point that the relay nodes are selected in a way that the CH may need to communicate directly with the fog node as long as its energy exceeds that in the relay node, causing a rapid node's death.

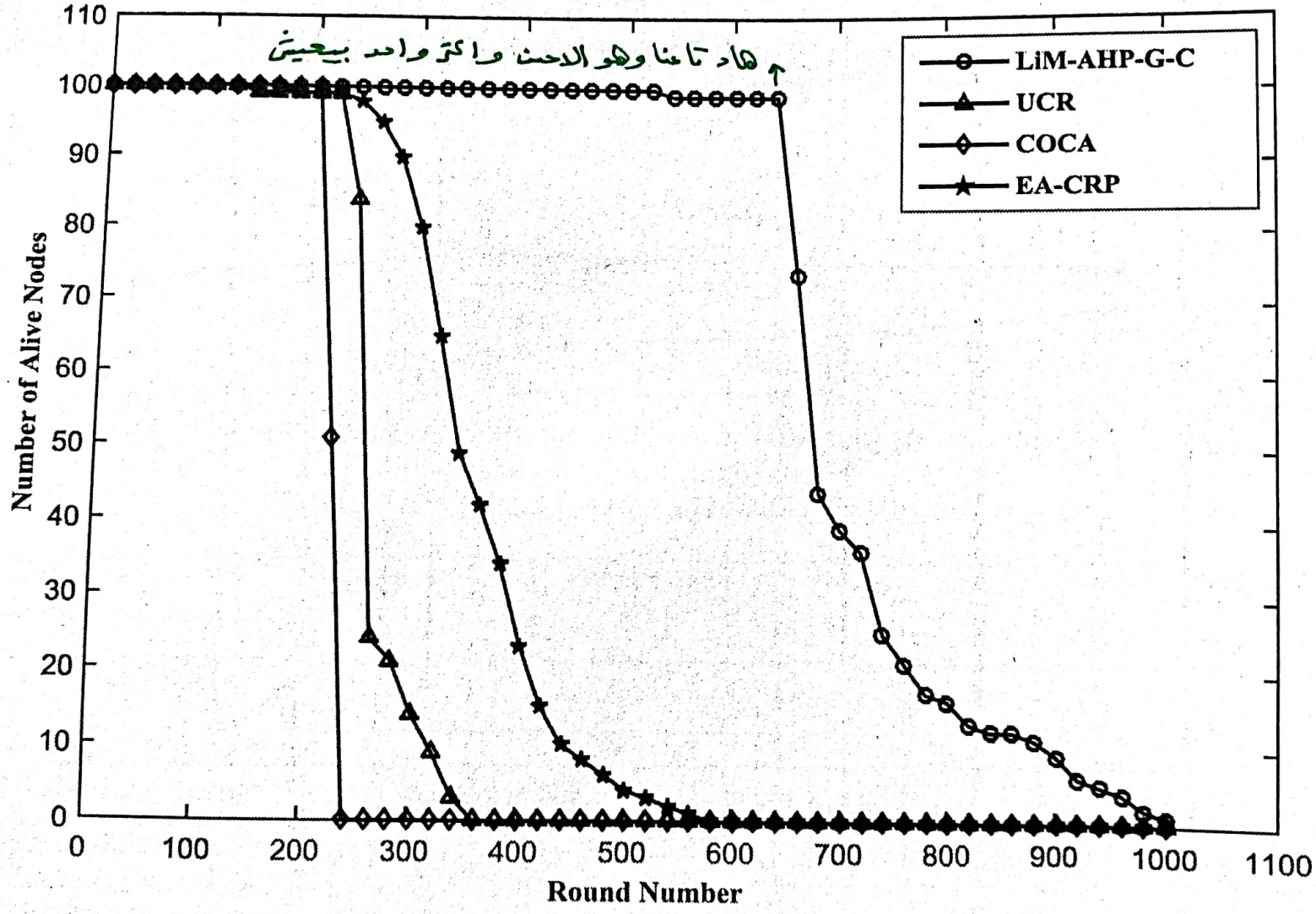


Figure 14. Number of alive sensors against round number considering different protocols

4.3.3 Simulation Scenario 3: FND Against Network Size

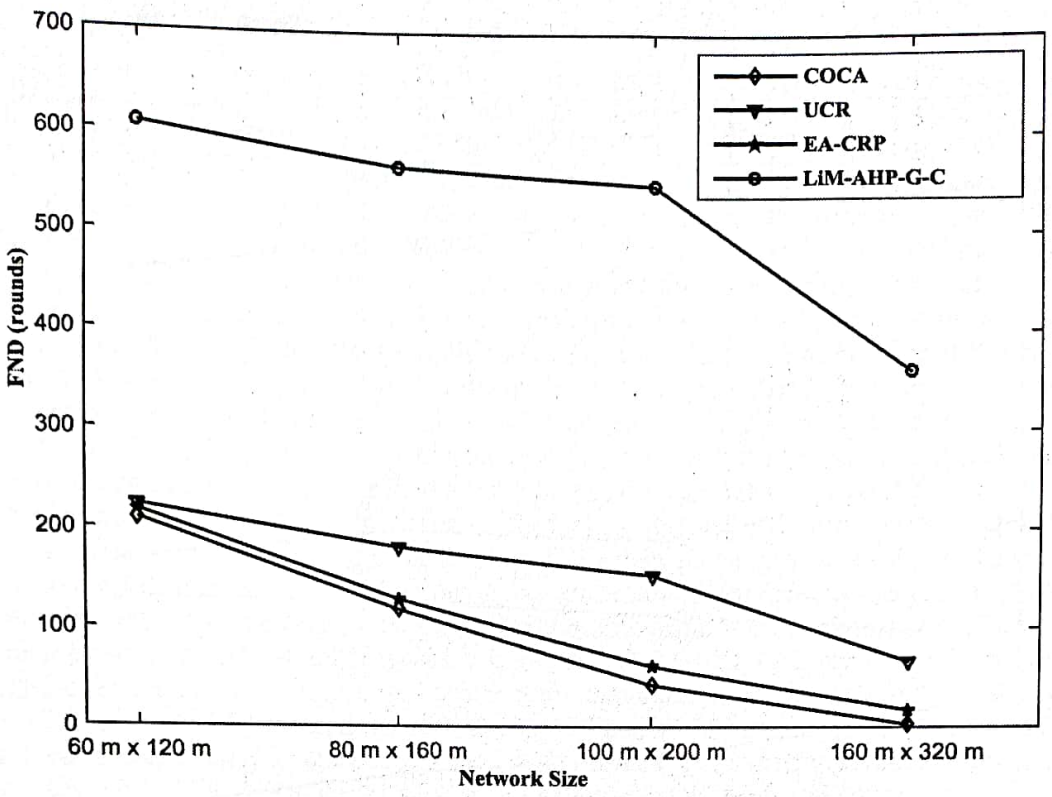


Figure 15. FND against sensor field size considering different protocols

4.3.5 Simulation Scenario 5: Network Energy Utilization Against Network Size

To this end, it is of interest to know the impact of the control overhead encountered in our proposed on the overall network energy utilization. In other words, what is the quota of the energy consumed for the control overhead with respect to the overall energy consumption. To address this concern, Fig. 17 depicts the network utilization in contrast with the network size considering all aforementioned protocols. The network utilization (NU) is defined as the proportion of the energy consumed in data transmission (E_D) to the total IoT network energy (E_T), which is expressed as:

$$NU = \frac{E_D}{E_T} \quad (19)$$

Among all protocols, our proposed protocol has the best network energy utilization. For instance and considering the network size of 60 m × 120 m, the network utilization in the LiM-AHP-G-C protocol approaches 98% while in the other protocols, that are, COCA, UCR, and EA-CRP, it approaches 33.5 %, 38.4%, and 49.5%, respectively. In a similar manner and considering the network size of 160 m × 320 m, it approaches 97%, 24%, 31.25% and 32.09% for LiM-AHP-G-C, COCA, UCR, and EA-CRP protocols, respectively. From these results, we can conclude that the control overhead in our proposed protocol is minimal. Furthermore, the scalability is achieved as the network utilization, in our proposed protocol, is much less sensitive to any change in the network size than others.

4.3.6 Simulation Scenario 6: Network Lifetime Opposite to Initial Energy

In this scenario and as shown in Fig. 18, the network lifespan, in terms of LND, opposite to the initial energy is examined considering our proposed protocol and its counterparts, that are, COCA, UCR, and EA-CRP. The values of initial energy are 0.5, 1, 1.5, 2, 2.5, and 3J. It can be clearly noticed from this figure that the proposed protocol surpasses the other protocols for all initial energy values considered. In addition, as the initial energy increases, the performance of all protocols gets better which is to be expected as the increase in the initial energy reflects positively on the IoT network lifetime. Unlike its counterparts, the behavior of our proposed protocol has a superliner trend as the initial energy increases. In fact, the ideas or algorithms incorporated in our proposed protocol make it novel and extremely attractive.

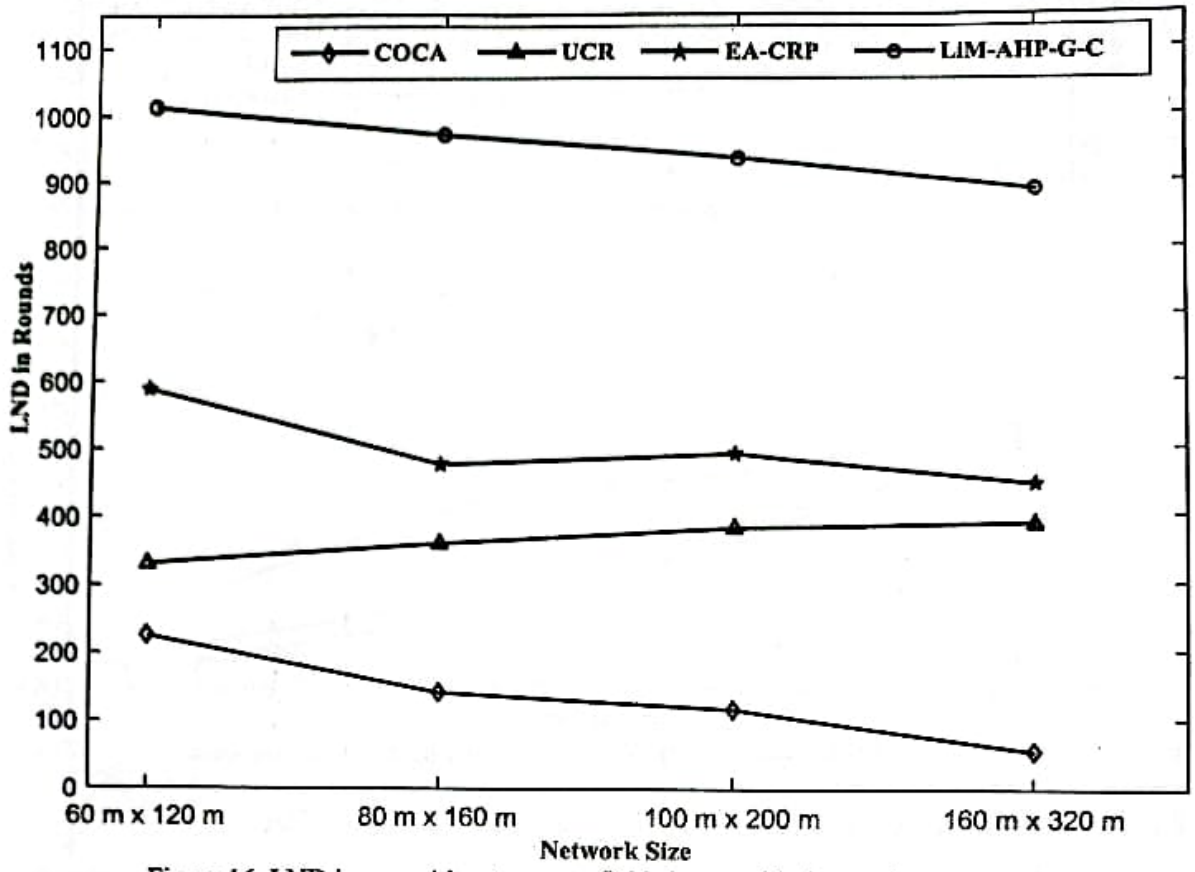


Figure 16. LND in opposition to sensor field size considering various protocols

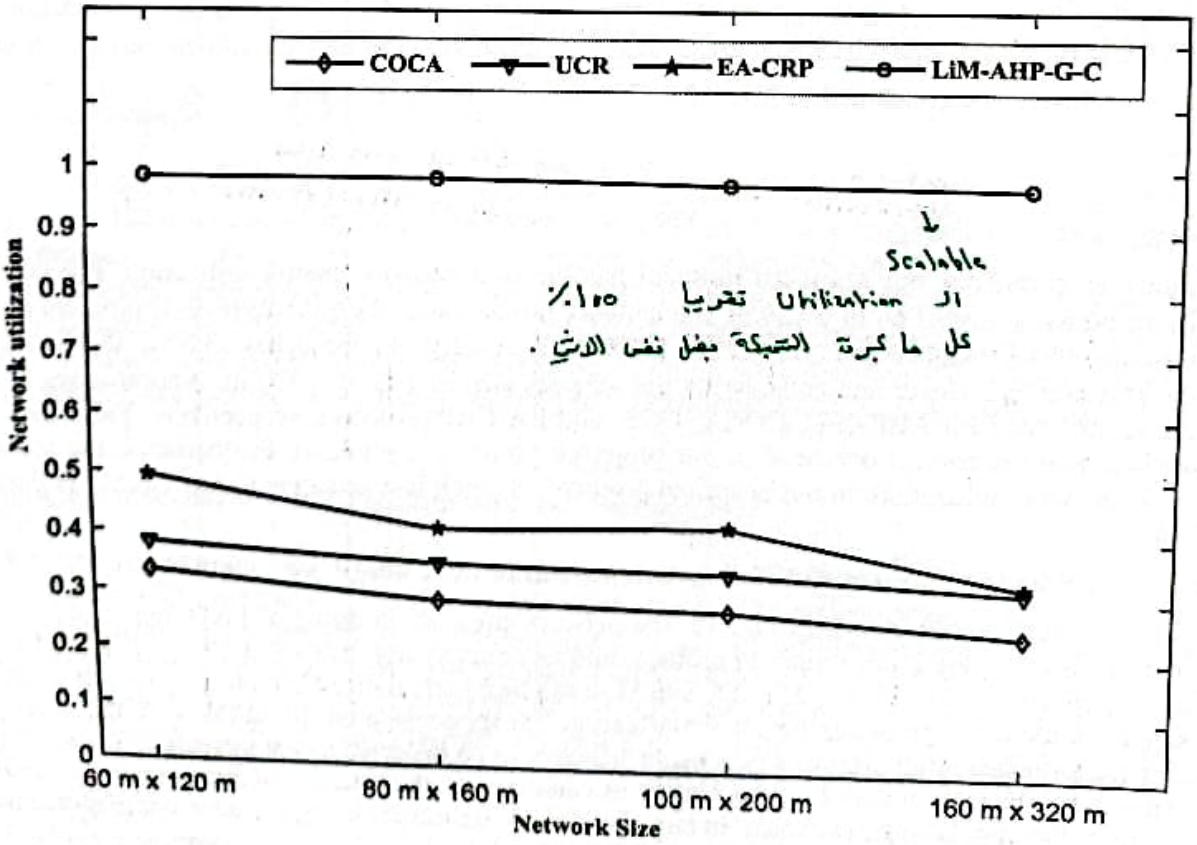


Figure 17. Network energy utilization against sensor field size considering various protocols

July 2019].

Appendix A

A.1 AHP Model Assumptions and Values

In this appendix, all assumptions and values belong to our AHP model are detailed. In particular, the matrix A, which is discussed in (5) and represents pairwise comparisons between the three dimensions (i.e., the levels of importance among the dimensions) chosen, is assumed, when referring to Table A.1, as

$$A = \begin{pmatrix} 1 & 3 & 5 \\ 0.333 & 1 & 3 \\ 0.2 & 0.333 & 1 \end{pmatrix}. \quad \text{(A.1)}$$

Utilizing (8), the A_{norm} can be written as

$$A_{norm} = \begin{pmatrix} 0.6523 & 0.6923 & 0.5556 \\ 0.2174 & 0.2308 & 0.3333 \\ 0.1304 & 0.0769 & 0.1111 \end{pmatrix}. \quad \text{(A.2)}$$

Utilizing (A.2) on (7) yields the local-weight vector (6) as

$$w = \begin{pmatrix} 0.6334 \\ 0.2605 \\ 0.1061 \end{pmatrix}. \quad \text{(A.3)}$$

Multiplying (A.1) by (A.3) results in obtaining the weight sum vector, w_s, as

$$w_s = \begin{pmatrix} 1.9454 \\ 0.7897 \\ 0.3195 \end{pmatrix}. \quad \text{(A.4)}$$

Substituting (A.3) and (A.4) into (12), we obtain

$$CO = \begin{pmatrix} 3.0714 \\ 3.0315 \\ 3.0113 \end{pmatrix}. \quad \text{(A.5)}$$

Substituting (A.5) into (11) yields

$$\lambda = 3.0381. \quad \text{(A.6)}$$

Substituting (A.6) into (10) provides

$$CI = 0.01904. \quad \text{(A.7)}$$

Consequently, substituting (A.7) into (9) and getting the *RI* from Table A.2, we obtain

$$CR = \frac{0.01904}{0.58} = 0.0328. \quad \text{(A.8)}$$

Referring to the consistency check constraint, illustrated previously, the one just obtained is within the proposed range. Therefore, our assumptions concerning the levels of importance among dimensions chosen are valid.

Table A.1. A pairwise comparison scale [41]

Scale of a_{ij}	Interpretation
1	Equally important.
3	Moderately important.
5	Strongly more important.
7	Very Strongly more important.
9	Extremely important.

Table A.2. RI based on the number of dimensions [41]

Number of dimensions (m)	RI
2	0
3	0.58
4	0.9
5	1.12
6	1.24
7	1.32
8	1.41
9	1.45
10	1.51

To find how the global-weight vector is obtained, let us assume that there is a cluster that consists of eight nodes where their IDs, residual energy, distance to the fog node, and average distance among other cluster members are all given in Table A.3.

Table A.3. Cluster members' energy and distances

Node ID	Residual energy (J)	Distance to the fog node (m)	Average distance among cluster members (m)
3	0.65	2.895	2.0234
5	0.559	1.956	1.0525
15	0.7	0.658	3.0652
8	0.61	0.9881	3.2081

53	0.5	1.0351	1.7110
91	0.57	3.0037	2.8198
96	0.79	2.9750	3.0022
100	0.68	1.0058	2.5581

With reference to the third step of section 3.5 and the values of the three dimensions of all cluster members, which are shown in Table A.3, the normalized version of the generated matrix B (i.e., B_{norm}) can be obtained as

Normalized → القيمة
Column مجموع ال

node 3
node 5
" 15

$$B_{norm} = \text{normalization of} \begin{pmatrix} 0.65 & 2.895 & 2.0234 \\ 0.559 & 1.965 & 1.0525 \\ 0.7 & 0.658 & 3.0652 \\ 0.61 & 0.9881 & 3.2081 \\ 0.5 & 1.0351 & 1.711 \\ 0.57 & 3.0037 & 2.8198 \\ 0.79 & 2.9750 & 3.0022 \\ 0.68 & 1.0058 & 2.5581 \end{pmatrix} = \begin{pmatrix} 0.128484 & 0.199302 & 0.104083 \\ 0.110496 & 0.135277 & 0.05414 \\ 0.138367 & 0.045299 & 0.157672 \\ 0.120577 & 0.068024 & 0.165023 \\ 0.098834 & 0.07126 & 0.088013 \\ 0.11267 & 0.206785 & 0.145049 \\ 0.156157 & 0.204809 & 0.154432 \\ 0.134414 & 0.069243 & 0.131587 \end{pmatrix} \quad \text{(A.9)}$$

Multiplying (A.9) by the local-weight vector, expressed in (A.3), results in having the following global-weight

بعد ما طلعنا ال Normalized
بضربها بال W

(8 x3) (3 x1)
= (8 x1)

$$W = \begin{pmatrix} 0.14434 \\ 0.11097 \\ 0.11617 \\ 0.11160 \\ 0.09050 \\ 0.14062 \\ 0.16864 \\ 0.11713 \end{pmatrix} \quad \text{(A.10)}$$

(Matrix ضرب) بين A.9 و W
كل عنصر element من العامود الصف بال Column كامل .

* لو حكانه ال Node اهم من ال Fog غالي اعلى الweight هي اي بتكون LH وايي بجدها CH

It is noteworthy to mention that the control packet received from the fog node includes primarily all clusters' members sorted in descending orders with reference to their AHP global weights. Based on this and referring to the global-weight vector of a cluster consists of eight nodes, as shown in (A.10), Table A.4 shows the portion of the control packet belongs to this cluster. Consequently, each cluster member will be aware of the heads in each batch. For example, nodes 96 and 3 will serve as CH and LH, respectively, in the current batch, while nodes 91 and 100 will serve as CH and LH, respectively, for the next batch and so forth.

لانه انا عايي انه Node مع ال Dfog
وال Dfog فيه اكثر أهمية . (ال Dfog بجدد ال CH وال Node بجدد LH)

* ال Setup اللي عنا معدودات .

Table A.4. Control packet portion belongs to eight nodes cluster

Node ID	3	5	15	8	53	91	96	100
Global weights	0.14434	0.11097	0.11617	0.11160	0.09050	0.14062	0.16864	0.11713
Node order	2	7	5	6	8	3	1	4

LH

برتيبهم بين الأكبر للاصغر

اعلى اشي
CH

E_{th}

بعض بضل هيك لحد مايصير اقل من الThreshold بعيد 91 و100 وهكذا

اعلى اشي CH ID : 96 (1)
LH ID : 3 (2)

بس اخلصه كامل بجمل Recluster من اول وجديد .

* اي عمل كل هاي الحسابات هوا Fog node وبعتت مسج كاملة لكل node مخلصات كاملة مين ال CH تابعه مين ال LH ويكون مرتبهم كاملين ، بس يخلصوا ببعتوا لل Fog انه خلصنا برجع يعمل Calculate وبجيد العملية

Appendix B

B.1 Selecting Optimal Relaying Nodes

To comprehend the operation of the genetic algorithm adopted in this paper, let us assume that we have a number of CHs (i.e., CH1, CH4, CH6, CH11, CH25, CH29, CH31), which require to transmit their data via optimal relay nodes to the fog node. Referring to this discussion of our genetic algorithm, the following steps are considered:

1) Each gene in the chromosome represents a CH along with its permissible relay node, as shown in Fig. B.1.

CH1	CH4	CH6	CH11	CH25	CH29	CH31
LH11	LH3	LH5	LH3	LH18	LH13	LH25

CH1	CH4	CH6	CH11	CH25	CH29	CH31
LH6	LH12	LH15	LH31	LH22	LH23	LH3

CH1	CH4	CH6	CH11	CH25	CH29	CH31
LH2	LH7	LH8	LH9	LH30	LH15	LH16

CH1	CH4	CH6	CH11	CH25	CH29	CH31
LH13	LH25	LH12	LH27	LH21	LH26	LH8

CH1	CH4	CH6	CH11	CH25	CH29	CH31
LH20	LH10	LH29	LH13	LH14	LH9	LH27

Figure B.1. Initial population

2) Selects the fittest chromosomes from the initial population (i.e., selects the chromosomes where the averages of their genes' fitness values stick within the boundaries). Figure B.2 shows the case where just four chromosomes are selected.

CH1	CH4	CH6	CH11	CH25	CH29	CH31
LH11	LH3	LH5	LH3	LH18	LH13	LH25

CH1	CH4	CH6	CH11	CH25	CH29	CH31
LH6	LH12	LH15	LH31	LH22	LH23	LH3

CH1	CH4	CH6	CH11	CH25	CH29	CH31
LH2	LH7	LH8	LH9	LH30	LH15	LH16

CH1	CH4	CH6	CH11	CH25	CH29	CH31
LH13	LH25	LH12	LH27	LH21	LH26	LH8

Figure B.2. Fittest chromosomes

3) Conducts a crossover function, based on the discrete technique, between the fittest parents where, for example, the pattern of vector V [0 1 1 0 101] is chosen. Simply, the genes of the new children will be taken from the first and second parents if the values of the corresponding vector's index are 0 and 1, respectively, as shown in Fig. B.3.

CH1	CH4	CH6	CH11	CH25	CH29	CH31
LH11	LH7	LH8	LH3	LH30	LH13	LH16

CH1	CH4	CH6	CH11	CH25	CH29	CH31
LH6	LH25	LH12	LH31	LH21	LH23	LH8

Figure B.3. New children after performing a crossover among parents

4) Performs a mutation function, utilizing the random resetting mutation method, among the children where, for example, the 4th gene of each new child (chromosome) is mutated based on the list {LH8, LH25, LH27, LH9} taking into consideration the selection of one entry at a time in order, as shown in Fig. B.4.

بعمل تزاوج كمان مرة
على نفس ال Pattern vector

CH1	CH4	CH6	CH11	CH25	CH29	CH31
LH11	LH7	LH8	LH8	LH30	LH13	LH16

CH1	CH4	CH6	CH11	CH25	CH29	CH31
LH6	LH25	LH12	LH25	LH21	LH23	LH8

Figure B.4. New children after performing a mutation among children

5) Conducts another crossover function between the parents which is similar to the that performed earlier, as shown in Fig. B.5.

CH1	CH4	CH6	CH11	CH25	CH29	CH31
LH11 ٥	LH25 ١	LH12 ١	LH8 ٥	LH21 ١	LH13 ٥	LH8 ١

Figure B.5. New child after conducting the second crossover

6) Obtains the optimal relay node for each CH after performing genes' mutation, similar to that performed in step 4, as shown in Fig. B. 6.

* العلامة بال 4th gene
اخدت 8 و 25 قبل هلأ بجمد
ال LH27

CH1	CH4	CH6	CH11	CH25	CH29	CH31
LH11	LH25	LH12	LH27	LH21	LH13	LH8

Figure B.6. Optimal relay nodes for CHs or CLs

* هيلاء بكون طلعت ال LH لكل CH

$$A = \frac{2\pi}{C_n}, \tag{1}$$

where C_n is the number of clusters in the network.

3.4 Sink Movement

In this work, the MS is proposed to move in a circular pattern with radius R_{ms} and a constant angular velocity, noting that R_{ms} is changed every number of rounds i depending on a quarter addition of the network radius to its prior value. After reaching three quarters of the network radius, R_{ms} will reversely follow the same forward trend. It is noteworthy to mention that the MS firstly broadcasts its initial position and angular velocity which will make it possible for each sensor node to predict the sink's instantaneous position at any time. It is known that the angular velocity $(\omega, \text{radians/sec})$ is identified as the ratio of the change in angular rotation $(\Delta\theta, \text{radians})$ to the change in time $(\Delta t, \text{sec})$. To have it clearer, the new MS location, which is, for example, at position k (P_{ms_k}) is being calculated as follow:

$$\theta_{1_{k-1}} = tan^{-1}\left(P_{ms_{k-1}} / P_{ms_0}\right) = tan^{-1}\left(\frac{y_{k-1} - y_0}{x_{k-1} - x_0}\right), \tag{2}$$

$$\theta_{2_k} = \theta_{1_{k-1}} + RT \times \omega, \tag{3}$$

$$P_{ms_k} \rightarrow \left(P_{ms_{k_x}}, P_{ms_{k_y}}\right) = R_{ms} \times \left(cos\left(\theta_{2_k}\right), sin\left(\theta_{2_k}\right)\right) \tag{4}$$

where, P_{ms_0} refers to the initial position of MS, $P_{ms_{k-1}}$ denotes for the last (previous) position of MS, and RT represents the round time. At equation (4) and to use degrees instead of radians, we consider this known conversion, namely, $360^o = 2\pi \text{ rad}$ or $1 \text{ rad} = 180^o / \pi$. To get started with this, we can initially assume, for instance, that the ratio at equation (2) is 1. Hence, the result of equation (2) will be 45^0 or $\left(\frac{45^0 \times 2\pi \text{ rad}}{360^0} = \frac{2\pi \text{ rad}}{8} = \frac{\pi \text{ rad}}{4}\right)$.

3.5 CH Selection using PSO

In this part, the process of selecting the CHs is described as revealed in Figure 4. CH nodes aggregate the received data from member nodes then forward them to the MS. As the nodes are randomly distributed, the selection of the CHs is essential for minimizing the energy consumption [41]. The selection of CHs is more demandable in homogeneous networks due the similarity in power for all nodes [42]. In this work, we use the PSO to find the best CH for each cluster.

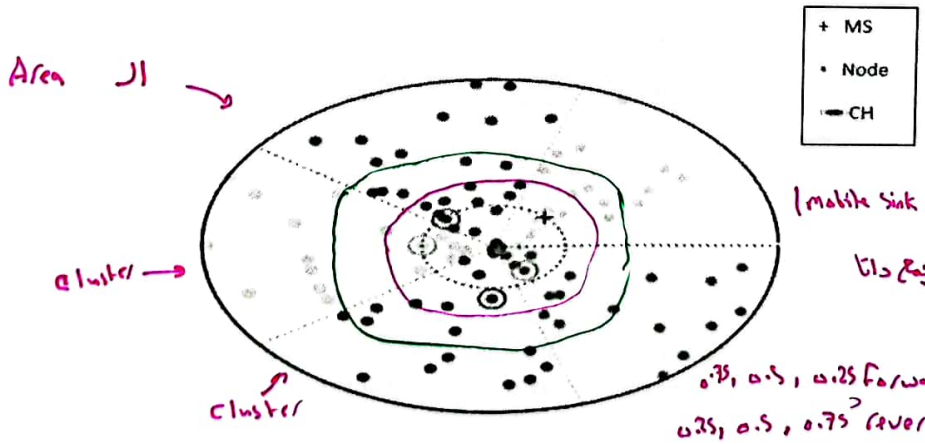


Figure 4. CH selection

* هون الـ PSO

It is worth stating that the MS disseminates a HELLO message for all sensor nodes. This message involves the related information of the MS, including the initial coordinate and the speed of the MS. Then, the nodes response with a HELLO message contains the node's ID, the node's coordinates and the node's energy. Based on that, the MS will have information about all nodes and further all nodes will have information about the MS. After that and as mentioned earlier, the MS starts the set-up state by dividing the network area into equal-sized clusters, distributing nodes among them and the CHs get elected based on PSO algorithm.

PSO [43] is a highly reputable optimization method and, in this work, used for selecting the most suitable node in a cluster to operate as a CH. In fact, the PSO can facilitate the implementation, control parameters, and apply more computational efficiency comparing with the other optimization algorithms. When initiating the process of PSO, which is an iterative approach, each particle is deployed with a random position and velocity. In each iteration, the particle finds its own personal best that is named "$P_{P_{best}}$" and also the global best that is named "G_{best}". To achieve the G_{best} solution, it uses its $P_{P_{best}}$ and G_{best} with the objective of updating the position and velocity.

In our work and as mentioned earlier, the PSO algorithm aims to select the best CH in each cluster, as a sufficient method to save energy and so to prolong the network lifetime. Thus, the distance from the nodes to the MS and the residual energy of the nodes are taking into consideration. It is assumed that fitness function (F) represents the residual energy of the sensor node along with the distance between the sensor node and the MS. Technically speaking, the PSO contains a predetermined number of particles (N_p) named a swarm, as particle gives a potential solution. Each particle P_i has two parameters, including its position and its velocity, where the fitness function (F) evaluates the quality of solution provided by each particle. The primary principle of PSO is to find the best particle's positions with the best result of fitness function (F). In more details, assume that f_1 is a function that expresses the distance between a sensor node and the MS $(D_{n\to ms})$. We need to maximize f_1 for optimal CH selection, thereby the distance between a node and the MS is recommended to be as short as possible. Also, we assume that f_2 is a function which basically refers to the ratio of node's residual energy (E_R) to the node's initial energy (E_{I_o}) bearing in mind that this ratio should be maximized to achieve the optimum CH election. The two objective functions (f_1 , f_2) are normalized between the range of 0 and 1. Noting that, these two functions will be used to find the fitness function (F) in the PSO and the appropriate way is to maximize their linear combination. Therefore, the optimal linear combination for CH selection is as follows:

Fitness ← $F = \alpha \times f_1 + (1-\alpha)\times f_2,$ (5)

where, ← $f_1 = 1-(D_{n\to ms}/R), \quad 0<f_1<1.$ (6)

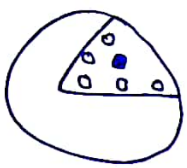
← $f_2 = E_R / E_{I_o}, \quad 0<f_2<1.$ (7)

where R refers to the radius of the network and α denotes for a parameter that normalizes the combination of f_1 and f_2 subject to $0<\alpha<1$.

PSO algorithm is repetitive, where in the initialization stage of PSO, the nodes in each cluster gets arranged in a matrix contains the ID, position and the energy of the nodes. Then the fitness function (F) is computed based on the matrix values. The node that has the maximum fitness function (F) is considered as the G_{best}. Thereafter, the algorithm forms a fitness matrix of all nodes in the cluster. Noting that, the particles are initialized by allocating random coordinates to the nodes. Every particle announces itself as a local or $P_{P_{best}}$ solution, and then it searches for the G_{best} solution. To find the G_{best} solution, during each iteration, every particle uses its own $P_{P_{best}}$ and G_{best} solution to update its position and velocity in order to reach the G_{best} solution, according to:

$$V_x^{(t+1)} = \omega \times V_x^{(t)} + c_1 \times r_1 \times \left(P_{best_x}^{(t)} - P_x^{(t)}\right) + c_2 \times r_2 \times \left(G_{best_x}^{(t)} - P_x^{(t)}\right) \quad (8)$$

* الـ PSO بقسم الـ nodes الـ عندي لـ 20 مثلا Particales
وبفلهم يتحركوا بدي اطلع اقل F (طلع اعلى Energy واقل Distance)
هدول الـ 20 بفلهم يلفوا بكل نقطة.
عشان يعرفوا مين افضل مكان وبعد عدد كبير من الـ rounds
بجد مين الأفضل.



Appendix A

Appendix A: CH Selection using PSO Example

In the initialization process of PSO, the nodes in each cluster get arranged in a matrix that contains the ID, the initial position, the initial velocity, and the energy of the nodes. Every particle announces itself as a P_{Pbest}, and then it searches for the G_{best} solution by calculating the fitness function (F), as mentioned previously in equations 5-7. For example, in cluster 1, as shown in Figure A.1, there are groups of 20 nodes. Therefore, we use the predefined swarm size of 20 particles, and the MS is situated in the position of (25, 25). Moreover, Table A.2 shows all nodes' details, including their ID, position, residual energy, initial velocity, initial P_{Pbest}, and calculated fitness based on equations 5-7. However, Table A.1 represents the main parameters utilized in the PSO equations, bearing in mind that these parameters have been selected after conducting a massive number of simulations and showed a noticeable and remarkable network performance enhancement.

For clarity, suppose we compute the fitness function (F) of the first node, whose position is (5.1, 9.36), then f_1 is 0.373, and f_2 is 0.26, referring to equations 6 and 7, respectively, detailed in subsection 3.5. Accordingly, and based on equation 5, $F_1^1 = 0.633$. The fitness of the remaining nodes is determined in a way that resembles the calculation of the first node's fitness, as shown in Table A.2. Afterward, the G_{best} solution is found based on the maximum fitness value (i.e., 0.639) between all nodes, in that case, node 13 is considered the initial G_{best} (i.e., $G^1_{best_x}$= 7.6865, $G^1_{best_y}$= 16.48).

Table A. 1. Equations parameters

Symbol	Definition	Value
(P_x,P_y)	MS position	(25,25)
R	Radius of the network	100
α	Normalization parameter	0.5
ω	Inertia weight	0.2
c_1	Acceleration coefficient	0.4
c_2	Acceleration coefficient	0.4
r_1, r_2	Random variables	Generated randomly

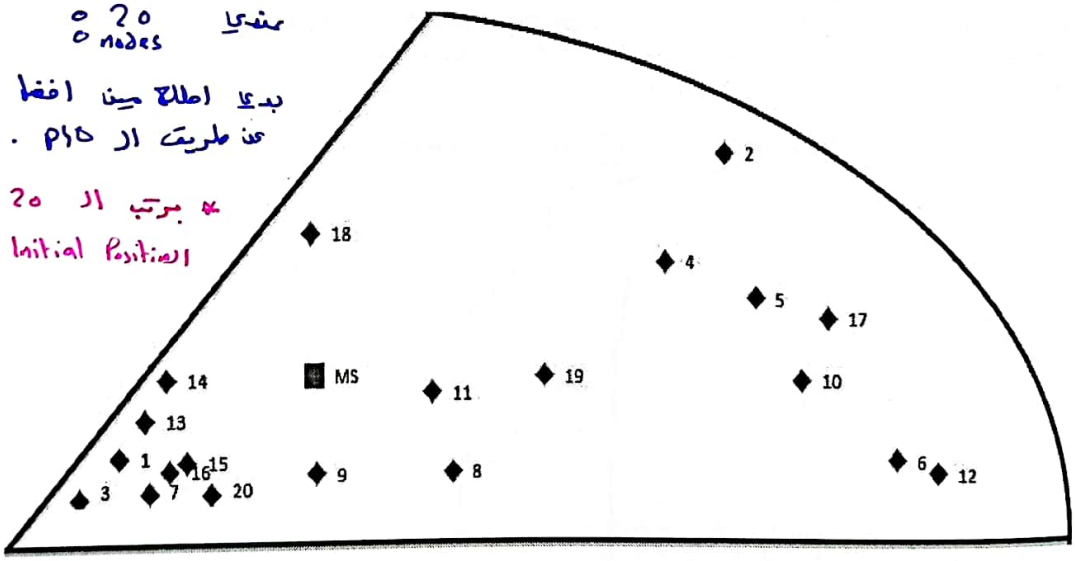


Figure A.1. Nodes distribution for cluster 1

Table A. 2. Initial cluster nodes' details

Node ID	Initial positions		E_R/E_{I_0}	Initial velocity		Personal best		Fitness
	P_x^1	P_y^1		V_x^1	V_y^1	$P^1_{x\_Pbest}$	$P^1_{y\_Pbest}$	F^1
1	5.1	9.36	0.52	0	0	5.1	9.36	0.633
2	68.26	67.76	0.37	0	0	68.26	67.76	0.381

3	1.01	1.5	0.34	0	0	1.01	1.5	0.502
4	61.92	46.58	0.37	0	0	61.92	46.58	0.471
5	71.53	39.53	0.32	0	0	71.53	39.53	0.416
6	86.47	8.28	0.29	0	0	86.47	8.28	0.326
7	4.06	2.5	0.4	0	0	4.06	2.5	0.546
8	39.49	6.17	0.21	0	0	39.49	6.17	0.486
9	25.29	5.95	0.42	0	0	25.29	5.95	0.615
10	76.39	23.76	0.28	0	0	76.39	23.76	0.383
11	37.3452	21.59	0.32	0	0	37.3452	21.59	0.596
12	90.8875	5.67	0.38	0	0	90.8875	5.67	0.347
13	7.6865	16.48	0.47	0	0	7.6865	16.48	0.639
14	9.8717	24.46	0.28	0	0	9.8717	24.46	0.564
15	11.8718	8.41	0.19	0	0	11.8718	8.41	0.489
16	11.1152	7.84	0.26	0	0	11.1152	7.84	0.520
17	79.2177	35.79	0.43	0	0	79.2177	35.79	0.439
18	24.6791	52.45	0.43	0	0	24.6791	52.45	0.578
19	49.1721	24.6	0.44	0	0	49.1721	24.6	0.599
20	14.3542	2.01	0.38	0	0	14.3542	2.01	0.563

As soon as the initialization phase ends, the PSO forms a fitness matrix of all nodes in the cluster at each iteration. Noting that, the particles are initialized by assigning coordinates to the nodes. The fitness matrix consists of nineteen columns as follows: *F(col. 1)* is the ID of the node, *F(col. 2)* and *F(col. 3)* are the current positions of a node on the x-axis (P_x^i), and y-axis (P_y^i), in sequence. *F(col. 4)* indicates the ratio of the residual energy of a node to its initial energy. However, the current velocity of a node on the x-axis (V_x^i), and y-axis (V_y^i) are represented in *F(col. 5)* and *F(col. 6)*, in a row. The $P^i_{x\_Pbest}$ and $P^i_{y\_Pbest}$ are the location of the current $P_{P_{best}}$ solution of a node, and they are shown in *F(col. 7)* and *F(col. 8)*. Interestingly, *F(col. 9)* is the fitness of the $P_{P_{best}}$ solution proposed by the ID^{th} node. *F(col. 10)* and *F(col. 11)* represent two generated random variables (r_1,r_2). *F(col. 12)* and *F(col. 13)*, which can be computed referring to equations A.1 and A.2, are the updated velocities for the next iteration on x-axis V_x^{i+1} and y-axis (V_y^{i+1}). However, the updated positions for the next iteration on the x-axis P_x^{i+1} and y-axis (P_y^{i+1}) are expressed in *F(col. 14)* and *F(col. 15)*, which can be computed referring to equations A.3 and A.4. *F(col. 16)* is the computed fineness after update the positions of a node. In addition to that, *F(col. 17)* and *F(col. 18)* represent the updated $P_{P_{best}}$ solution after the comparison is made. In other words, if the $F^i(Pbest) \geq F^{i+1}(Pbest)$, then the $F^i(Pbest)$ remains the same without change, otherwise, $F^i(Pbest)$ is updated to become the value of $F^{i+1}(Pbest)$, which is clearly denoted in *F(col. 19)*, and then the algorithm searches now for the G_{best}. It is worth mentioning that, in general, to find the G_{best} solution (i.e., the maximum value of *F(col. 19)*), during each iteration, every particle uses its own $P_{P_{best}}$ and G_{best} solution to update its position and velocity in order to reach the G_{best}, which are detailed in equations A.1-A.4, shown below:

$$F(col.\ 12)=\omega \times F(col.\ 5)+c_1 \times F(col.\ 10)\times(F(col.7)-F(col.\ 2))+c_2 \times F(col.11)\times(G_{best_{x(i)}}-F(col.\ 2)) \quad (A.1)$$

$$F(col.\ 13)=\omega \times F(col.\ 6)+c_1 \times F(col.\ 10)\times(F(col.8)-F(col.\ 3))+c_2 \times F(col.11)\times(G_{best_{x(i)}}-F(col.\ 3)) \quad (A.2)$$

$$F(col.\ 14)=F(col.\ 2)+F(col.\ 12) \quad (A.3)$$

$$F(col.\ 15)=F(col.\ 3)+F(col.\ 13) \quad (A.4)$$

- **Iteration 1:**

To this point, we will start iteration 1, where the concept of particles arises. Every particle considers itself as the best personal solution and then it searches for the G_{best} solution. It is noteworthy to mention that a new velocity and position for each particle is found for each iteration based on equations A.1-A.4. As a result, Table A.3 shows the fitness matrix at iteration 1. Referring to the entries of Table A.3, every node updates its velocities and positions based on the previous velocities and positions. After that, the new fitness is computed. If the new fitness is lower than (or the same as) of the fitness of the $P_{P_{best}}$ solution (i.e., $F^1(P_{P_{best}})$, then there will be no change in this solution. Particularly, the P_{Pbest} will remain as is (like the nodes 13 and 14). On the other hand, if the new computed fitness of

Table A. 3. Updated fitness matrix at iteration 1

Node ID	Initial positions		E_R/E_{I_0}	Initial velocity		Personal best		Initial $F^1(P_{P_{best}})$	Random numbers		Updated velocities		Updated positions		Updated fitness	Updated personal best		$F(Pbest)$
	P_x^1	P_y^1		V_x^1	V_y^1	$P_{x\_Pbest}^1$	$P_{y\_Pbest}^1$	F^1	r_1	r_2	V_x^2	V_y^2	P_x^2	P_x^2	F^2	$P_{x\_Pbest}^2$	$P_{y\_Pbest}^2$	
1	5.1	9.36	0.52	0	0	5.1	9.36	0.633	0.914	0.414	0.428	1.178	5.528	10.538	0.639	5.528	10.538	0.639
2	68.26	67.76	0.37	0	0	68.26	67.76	0.381	0.275	0.785	-19.018	-16.100	49.242	51.660	0.505	49.242	51.660	0.505
3	1.01	1.5	0.34	0	0	1.01	1.5	0.502	0.279	0.373	0.997	2.237	2.007	3.737	0.513	2.007	3.737	0.513
4	61.92	46.58	0.37	0	0	61.92	46.58	0.471	0.649	0.613	-13.299	-7.381	48.621	39.199	0.547	48.621	39.199	0.547
5	71.53	39.53	0.32	0	0	71.53	39.53	0.416	0.210	0.674	-17.210	-6.213	54.320	33.317	0.508	54.320	33.317	0.508
6	86.47	8.28	0.29	0	0	86.47	8.28	0.326	0.038	0.148	-4.673	0.486	81.797	8.766	0.350	81.797	8.766	0.350
7	4.06	2.5	0.4	0	0	4.06	2.5	0.546	0.579	0.200	0.290	1.119	4.350	3.619	0.551	4.350	3.619	0.551
8	39.49	6.17	0.21	0	0	39.49	6.17	0.486	0.968	0.578	-7.350	2.383	32.140	8.553	0.515	32.140	8.553	0.515
9	25.29	5.95	0.42	0	0	25.29	5.95	0.615	0.964	0.932	-6.562	3.925	18.728	9.875	0.628	18.728	9.875	0.628
10	76.39	23.76	0.28	0	0	76.39	23.76	0.383	0.508	0.980	-26.943	-2.855	49.447	20.905	0.516	49.447	20.905	0.516
11	37.3452	21.59	0.32	0	0	37.3452	21.59	0.596	0.820	0.199	-2.364	-0.407	34.982	21.183	0.607	34.982	21.183	0.607
12	90.8875	5.67	0.38	0	0	90.8875	5.67	0.347	0.377	0.490	-16.300	2.118	74.588	7.788	0.428	74.588	7.788	0.428
13	7.6865	16.48	0.47	0	0	7.6865	16.48	0.639	0.353	0.751	0.000	0.000	7.687	16.480	0.639	7.687	16.480	0.639
14	9.8717	24.46	0.28	0	0	9.8717	24.46	0.564	0.986	0.735	-0.643	-2.347	9.229	22.113	0.560	9.872	24.460	0.564
15	11.8718	8.41	0.19	0	0	11.8718	8.41	0.489	0.095	0.814	-1.362	2.627	10.509	11.037	0.494	10.509	11.037	0.494
16	11.1152	7.84	0.26	0	0	11.1152	7.84	0.520	0.865	0.841	-1.154	2.908	9.961	10.748	0.526	9.961	10.748	0.526
17	79.2177	35.79	0.43	0	0	79.2177	35.79	0.439	0.048	0.191	-5.471	-1.477	73.747	34.313	0.467	73.747	34.313	0.467
18	24.6791	52.45	0.43	0	0	24.6791	52.45	0.578	0.039	0.510	-3.467	-7.339	21.212	45.111	0.613	21.212	45.111	0.613
19	49.1721	24.6	0.44	0	0	49.1721	24.6	0.599	0.656	0.773	-12.825	-2.510	36.347	22.090	0.661	36.347	22.090	0.661
20	14.3542	2.01	0.38	0	0	14.3542	2.01	0.563	0.235	0.043	-0.115	0.251	14.239	2.261	0.564	14.239	2.261	0.564

Fitness اعلى global best

هو بيعطينا اياهم .

صار ن Velocity معادلة (A.1, A.2)

معادلة (A.3, A.4)

f2 , f1 ← لل Particle والـ (MS)

* بشوف مين اعلى ال node ولد ال Particle وهو اي باخذه .

* هاد مش global هاد اعلاهو Personal لانه لكل واحد . زي بين 0.639 و 0.633

* نتائج iteration 1 باقي كذا وهكذا .

* لقينا 100 iteration لحد ما ثبتت الارقام .

* مهما لف يقلد إلا هذا الاحسن بالنسبة للـ MS بعد ما جربنا 100 iteration وضل gbest

* مهما تغيرت الوضعيات .

Table A. 4. Updated fitness matrix at iteration 2

Node ID	Current positions		E_R/E_{I_0}	Current velocity		Current personal best		$F^2(P_{P_{be}}$	Random numbers		Updated velocities		Updated positions		Fitness After update	Updated personal best		$F(Pbest)$
	P_x^2	P_y^2		V_x^2	V_y^2	$P^2_{x\_Pbest}$	$P^2_{y\_Pbest}$	F^2	r_1	r_2	V_x^3	V_y^3	P_x^3	P_x^3	F^3	$P^3_{x\_Pbest}$	$P^3_{y\_Pbest}$	
1	5.528	10.538	0.52	0.428	1.178	5.528	10.538	0.639	0.469	0.732	9.104	3.616	14.632	14.154	0.685	14.632	14.154	0.685
2	49.242	51.660	0.37	-19.018	-16.100	49.242	51.660	0.505	0.909	0.250	-5.091	-6.173	44.151	45.487	0.545	44.151	45.487	0.545
3	2.007	3.737	0.34	0.997	2.237	2.007	3.737	0.513	0.253	0.352	5.033	3.031	7.040	6.768	0.542	7.040	6.768	0.542
4	48.621	39.199	0.37	-13.299	-7.381	48.621	39.199	0.547	0.119	0.358	-4.418	-3.927	44.204	35.272	0.576	44.204	35.272	0.576
5	54.320	33.317	0.32	-17.210	-6.213	54.320	33.317	0.508	0.492	0.690	-8.400	-4.340	45.920	28.977	0.554	45.920	28.977	0.554
6	81.797	8.766	0.29	-4.673	0.486	81.797	8.766	0.350	0.694	0.874	-16.816	4.753	64.982	13.519	0.437	64.982	13.519	0.437
7	4.350	3.619	0.4	0.290	1.119	4.350	3.619	0.551	0.069	0.806	10.378	6.181	14.728	9.800	0.608	14.728	9.800	0.608
8	32.140	8.553	0.21	-7.350	2.383	32.140	8.553	0.515	0.008	0.184	-1.160	1.474	30.980	10.027	0.524	30.980	10.027	0.524
9	18.728	9.875	0.42	-6.562	3.925	18.728	9.875	0.628	0.948	0.243	0.403	1.975	19.131	11.850	0.638	19.131	11.850	0.638
10	49.447	20.905	0.28	-26.943	-2.855	49.447	20.905	0.516	0.790	0.889	-10.046	-0.150	39.400	20.755	0.565	39.400	20.755	0.565
11	34.982	21.183	0.32	-2.364	-0.407	34.982	21.183	0.607	0.062	0.359	-0.277	0.049	34.705	21.231	0.608	34.705	21.231	0.608
12	74.588	7.788	0.38	-16.300	2.118	74.588	7.788	0.428	0.014	0.012	-3.450	0.495	71.138	8.282	0.445	71.138	8.282	0.445
13	7.687	16.480	0.47	0.000	0.000	7.687	16.480	0.639	0.739	0.848	9.717	1.902	17.403	18.382	0.685	17.403	18.382	0.685
14	9.229	22.113	0.28	-0.643	-2.347	9.872	24.460	0.564	0.231	0.371	3.953	-0.256	13.182	21.856	0.579	13.182	21.856	0.579
15	10.509	11.037	0.19	-1.362	2.627	10.509	11.037	0.494	0.848	0.214	1.941	1.472	12.451	12.509	0.506	12.451	12.509	0.506
16	9.961	10.748	0.26	-1.154	2.908	9.961	10.748	0.526	0.786	0.426	4.261	2.512	14.222	13.260	0.550	14.222	13.260	0.550
17	73.747	34.313	0.43	-5.471	-1.477	73.747	34.313	0.467	0.293	0.331	-6.040	-1.912	67.707	32.401	0.498	67.707	32.401	0.498
18	21.212	45.111	0.43	-3.467	-7.339	21.212	45.111	0.613	0.920	0.406	1.767	-5.210	22.979	39.901	0.640	22.979	39.901	0.640
19	36.347	22.090	0.44	-12.825	-2.510	36.347	22.090	0.661	0.033	0.974	-2.565	-0.502	33.782	21.588	0.673	33.782	21.588	0.673
20	14.239	2.261	0.38	-0.115	0.251	14.239	2.261	0.564	0.916	0.137	1.185	1.134	15.424	3.394	0.572	15.424	3.394	0.572

An Innovative RPL Objective Function for Broad Range of IoT Domains Utilizing **Fuzzy Logic** and **Multiple Metrics**

Khalid A. Darabkh[a*], Muna Al-Akhras[a], Ala' F. Khalifeh[b], Iyad F. Jafar[a], and Fahed Jubair[a],

[a]Department of Computer Engineering, The University of Jordan,

Amman, 11942, Jordan

[b]Department of Electrical Engineering, German Jordanian University, Amman 11180, Jordan

*Corresponding Authors Email: k.darabkeh@ju.edu.edu.jo

Abstract - Recently, the Internet of Things (IoT) is foreseen to be a significant part of the future Internet. The indispensable part of IoT is the Low-power and Lossy Network (LLN), which is required to connect plenty of resource-constrained (e.g., power and memory) wireless devices. As a result, routing protocols play a vital role in providing LLN components with the requisite interoperability capabilities. Because of this, the IETF announced that the IPv6 Routing Protocol for LLNs (i.e., RPL) is the standard routing protocol for IoT. Actually, RPL relies on constructing a destination-oriented directed acyclic graph governed by a variety of routing metrics that help in choosing the Objective Function (OF), which is liable for selecting the best preferred parent of each node which in turn gets involved in the route establishment to the destination. In RPL, two OFs are commonly used, namely, OF zero and minimum rank with hysteresis OF which use the metrics of hop count and expected transmission count, respectively. In fact, establishing an OF based on a single metric has flaws or drawbacks, as it only benefits a few IoT applications. To put it another way, the network lifetime is improved for a few IoT apps but degraded for the majority. Therefore, developing an OF with a composite metric, which performs well across a broad range of IoT domains, is very challenging and presently attracting the attention of IoT researchers. In this article, we address the challenge of using a single metric besides addressing the limitations of those works employed a composite metric by putting forward a cross-layer design and accordingly developing a fuzzy logic system that brings together four input metrics, namely, hop count, energy consumption, latency, and received signal strength indicator as a new OF, abbreviated as FL-HELR-OF. The simulation findings, acquired by the Cooja simulator, prove the effectiveness of our new OF, which particularly outperforms other existing studies concerning the packet delivery ratio, control message overhead, latency, energy consumption, and average hop count.

Keywords: IoT; LLN; RPL; OF; Fuzzy Logic; Efficient Energy; Network Lifetime

1. Introduction

Nowadays, the Internet of Things (IoT) has been the label of the new era of internet networking (Harb, et al., 2017). IoT is made up of a variety of physical devices, for example, sensors, vehicles, or even devices equipped with microprocessors (Al-Fuqaha, Guizani, Mohammadi, Aledhari, & Ayyash, 2015) (Darabkh, Zomot, Al-qudah, & Khalifeh, 2022). The smart cities applications, such as healthcare, military, waste management, air quality, noise monitoring, and traffic congestion, can all benefit from these devices (Kassab & Darabkh, 2020). This type of networking authorizes devices to collect and interchange a colossal amount of data (Darabkh & Al-Jdayeh, 2019). Additionally, the IoT network is connected to the real world via the integration with computer systems (Banh, et al., 2016). In (Evans, 2011), Cisco anticipated that there would be nearly 100 billion devices interconnected through

IoT by the year 2025 (Sturman, 2021). What is more, by 2025, Internet devices will have a substantial part in people's daily life (Darabkh, Kassab, & Khalifeh, 2020).

In developing markets, by way of designing and building effective and operable commercial devices, it is necessary to activate networks that are named Low-power and Lossy Network (LLN) (Janicijevic, Lukic, & Mezei, 2011). One of the paramount issues in LLNs is developing an efficient routing protocol that satisfies user and application requirements. That is due to the severe characteristics of the LLNs that have low-powered nodes and a short transmission range (Kharrufa, Al-Kashoash, & Kemp, 2019). The purposes of designing the Routing Protocol for Low Power and Lossy Networks (RPL) refer to the routing needs which were determined by IETF in 2009 through revealing application domains of building and home automation of urban and industrial networks. After several years, it is important to acknowledge that these conditions are fulfilled by RPL. Interestingly, RPL focuses on creating highly efficient and stable routes between one or more root nodes and all the other nodes. Actually, RPL is a proactive loop-free distance vector routing protocol (Kim, Ko, Culler, & Paek, 2017). RPL constructs a Destination-Oriented Directed Acyclic Graph (DODAG) based on **node and link metrics** such as link quality level and Hop Count (HC) (Expected Transmission Count (ETX)) remaining energy, or consumption energy (Darabkh & Al-Akhras, December 2021). These metrics are required to design an Objective Function (OF), which is responsible for selecting the preferred parent of a node (Capone, Brama, Accettura, Striccoli, & Boggia, 2014). As such, the nodes continue the process of choosing the optimum route until reaching the destination.

In real-world systems, there are many regions of ambiguity or fuzziness, and the most effective method to deal with this is to employ fuzzy logic, which is simple to implement and proven to be robust (Dimitroulis & Alamaniotis, 2022). Therefore, we use in this work a type-1 fuzzy set (Mamdani & Assilian, 1975) to implement a unique OF, taking into account that the exact membership function is decided by a variety of factors, including extensive simulations based on trial-and-error techniques, existing literature, device datasheet, and application requirements. Despite this, our improved version of RPL, which will be discussed shortly, is still susceptible to disturbances, modeling errors, and other uncertainties found in real-world applications. Surprisingly, there have been recent proposals or mechanisms for overcoming and effectively dealing with the aforementioned impairments where interested readers may refer to (Zhang, et al., 2021) (Zhang, He, Stojanovic, & Luan, 2021) (Xin, et al., 2021) (Xu, Li, & Stojanovic, 2021) to explore more details on this.

1.1 Problem Definition and Aims

As already pointed out, RPL uses OF to select the best path to a DODAG root, whereas there are two widely used OFs defined for the standard RPL, that are, (OF Zero (OF0) (Thubert, 2012) and (Minimum Rank with Hysteresis OF (MRHOF)) (Gnawali & Levis, 2012) which use HC and ETX as routing metrics, respectively. Actually, the OF0 and MRHOF decide the best path to the destination through the preferred parent of the node. In the case of OF0, the minimum hops are the OF. Even as the MRHOF, the minimum ETX count across the path is the OF. Several studies reported that these OFs have many obstacles due to high energy consumption and unreliability of the link (Liu, Sheng, Yin, Ali, & Roggen, 2017) (Kim, Ko, Culler, & Paek, 2017). Additionally, they did not meet the requirements of the majority of the IoT applications, thereby degrading the overall network performance) (Triantafyllou, Sarigiannidis, & Lagkas, 2018) (Kharrufa, Al-Kashoash, & Kemp, 2019). Using a single metric has proved incompetence in improving the performance of RPL (Ghaleb, et al., 2019) (Lamaazi & Benamar, 2020). It is worth noting that both aforementioned OFs utilize a single metric. To comply with these needs, the authors in (Kamgueu, Nataf, & Ndie Djotio, 2015) paved the way for using a composite metric (i.e., raw input metrics) instead of using one metric where these metrics can be combined to define a new OF. It is noteworthy to mention that RPL has no guidelines on how to implement such a combination.)

(Along these lines, RPL is considered user-dependent and must be consistent with the application requirements) (For instance, medical applications necessitate minimizing end-to-end delay and expanding the network lifetime) while some applications focus on having energy efficiency, and others are intended to attain reliability and so on. Nevertheless, we should make an adequate trade-off between the network aspects bearing in mind that the requirements of applications are frequently synchronized with each other, e.g., the enhancement in the network load balancing may influence, in a negative way, the end-to-end delay. In light of the above, our aim

ETX :

* قسيش عدد الـ trans. اللي بتوقع اللي الـ node بتوقع يصلهم distination لحد ما يوصل الـ Packet Succes . (بين الـ neighbors)

* اذا واحد عنده مشكلة رح يأثر على الـ ETX

$$\frac{1}{P_{Pack} * P_{Ack}}$$

P_{Pack} ← neighbor link quality (LQ)

P_{Ack} ← node link quality (تاني) (NLQ)

مثالة: Round trip يعني روحة ورجعة

متان اسب الـ total link quality بدي LQ X NLQ

لو كانت round trip prob. = %64 (حاصل ضرب الـ LQ , NLQ)

يعني الـ LQ = %80

NLQ = %80

يعني كل 100 packet عندي 64 packet رح يعملوا round trip .

$$\frac{1}{0.64} = 1.5625$$ ← round trip prob.

عدد المرات اللي لازم ابعت الـ packet متان توصل Succes.

عشي بدي امشي من A → B → C → D

ETX_1 ETX_2 ETX_3

$$ETX\ total = ETX_1 + ETX_2 + ETX_3$$

كلما زدنا عدد الـ neighbors اكتر بيصير الـ ETX معقد اكتر.

illustrates the proposed protocol extensively. The conducted simulation results are presented and discussed in Section 5. Finally, Section 6 concludes our work and provides some useful future directions.

2. Overview of RPL

This section discusses the motivation behind standardizing RPL, types of RPL control messages, RPL OFs, and RPL operation, in sequence.

With the exponential growth of the Internet and revolutionary of IoT, traditional routing protocols can no longer serve a large number of added devices in LLNs. Thus, the protocol stacks were stated by the IETF and IEEE by setting various layers' protocols, which are shown in Table 1 (Iova, Picco, Istomin, & Kiraly, 2016) (Sebastian & Sivagurunathan, 2018). One of the preeminent standards is the (IEEE 802.15.4) which represents the crux of IoT networks. It is used by LLNs and basically covers the network's PHY and data-link layers. For the purpose of integrating low-end devices with the Internet, the IETF has devised the IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) (Moritz & Lerche, 2013). This layer is considered as an adaptation layer to permit sensor nodes to activate the IP stack and make it accessible to all devices connected to the network. Furthermore, the adaptation layer makes it possible for the nodes to run routing protocols at the network layer and enables the launch of many applications by availing the end-to-end connectivity. Thus, the specifications of the 6LoWPAN promote the working teams of ROLL to standardize RPL in 2012 (Winter, et al., 2012). It is a proactive IPv6 distance-vector protocol designed for large networks with a huge number of active devices in dynamic and lossy link networks (Sharma & Shukla, 2014). Nevertheless, the objectives of designing the RPL are related to the routing needs determined by IETF in 2009 through reflecting the application domains of building and home automation (Brandt, Buron, & Porcu, 2010) of urban and industrial networks (Dohler, Watteyne, Winter, & Barthel, 2009), (Pister, Thubert, Dwars, & Phinney, 2009), (Gubbi, Buyya, Marusic, & Palaniswami, 2013). Three types of traffic flows are adopted by RPL, namely, Point to Point (P2P), Point to Multipoint (P2MP), and Multipoint to Point (MP2P) (Sobral, Rodrigues, Rabêlo, Al-Muhtadi, & Korotaev, 2019) (Iova, Picco, Istomin, & Kiraly, 2016). However, there are four types of RPL control messages which allow exchanging data between nodes in the DODAG and are detailed as follows (Winter, et al., 2012):

- (DODAG) Information Solicitation (DIS) which is used to inquire a RPL node for a DIO. In particular, the DIS message triggers the transmission of DIO since the node, that wants to join the DODAG, sends a DIS message to the root, which in turn sends a DIO message that contains all information of this DODAG.
- DIO, which is considered the most momentous control message in RPL and specifically used to discover and maintain upward routes, which will be detailed shortly. It is initiated by a DODAG root and retransmitted in a broadcast style (by its neighbors using the trickle timer algorithm). Also, the DIO message chooses a DODAG parent set and maintains the DODAG consistency. It carries information that permits a node to explore a RPL instance and retrieves its configuration parameters, out of which instance ID, version number, current rank, routing metrics, Objective Code Point (OCP), DODAG ID.
- DODAG Advertisement Object (DAO), which is used to construct and maintain the downward route, which will be thoroughly explained quite soon. It is further utilized to broadcast the destination information from the leaf (child) node upwards toward a DODAG root. In the non-storing mode, the DAO message will be transmitted from a leaf node to a DODAG root. In contrast, the child node in the storing mode will send a DAO message to the selected parent(s)/next hop instead.
- DODAG advertisement object acknowledgment, which is transmitted as a unicast packet by a DAO receiver (a DAO parent or DODAG root) in response to a unicast DAO message.

The RPL OF identifies how the nodes select the routes along the DODAG. Besides, it enables the nodes to divert a set of metrics and constraints into a single value named "*rank*", which specifies how far a node is from a DODAG root (i.e., the closer, the lower). Interestingly, the rank of a node is mainly computed by adding the rank of the preferred parent to the rank-increase or step. Moreover, the OF assists nodes in determining their parents. This can be accomplished by configuring the DIO messages, which enclose the advertised metrics and constraints therein (Thubert, 2012). Each RPL instance is associated with a specified OF.

Table 1: IETF standardization for the IPv6 protocol stacks

Layer	Protocol (s)
Application Layer	CoAP
Transport Layer	UDP
Network Layer	IPv6 ICMP IETF RPL
Adaption Layer	6LoWPAN
MAC Layer	IEEE 802.15.4 MAC
PHY Layer	IEEE 802.15.4 PHY

2.1 RPL Operations

The DODAG has one path from each single leaf node to the root. This is achieved by selecting the appropriate OF, constraints, and metrics to ultimately define the routing topology. As soon as the DODAG is built, there are many roles that nodes may act in the network: 1) Root nodes, which are the roots of the graph and particularly those connected to the Internet or any outer infrastructure, 2) Router nodes, which are located between roots and leaf nodes and particularly used to forward traffic towards the roots, and 3) Leaf nodes, which are placed at the edges of the graph and cannot be used to send or forward traffic. RPL has a proactive route discovery and topology construction mechanism by transmitting a periodic DIO message, which is controlled by the trickle timer algorithm that manages its transmission rate (Levis, Clausen, Hui, Gnawali, & Ko, 2011). The DODAG's construction is initiated by the root node. The construction phase can be done in both directions, upward and downward routes, as follows:

- **Upward Route:** In RPL and for data transmission purposes, a graph (DODAG) is created to find the best path from each node to a DODAG root. Referring to the DIO format, discussed earlier, it is important to remember that the PRL instance ID is a network-wide identifier bearing in mind that DODAGs with the same RPL instance ID have the same OF. The RPL instance, on the other hand, is a collection of one or more DODAGs that have a RPL instance ID. In a RPL instance, a RPL node can only belong to one DODAG. In fact, each RPL instance runs on its own and is unaffected by the actions of other RPL instances. A DODAG root's identification is called a DODAG ID which is unique for every RPL instance. Importantly, a DODAG is identified by the entry (RPL instance ID, DODAG ID). However, a DODAG version number is somehow a counter that is increased by the root to generate a new version of a DODAG. As a result, a DODAG version is a certain iteration ("*version*") of a DODAG with a specific DODAG ID. Essentially, the entry (RPL instance ID, DODAG ID, DODAG version number) distinctively identifies a DODAG version. Initially, a DODAG root broadcasts a DIO message to its neighbors which in turn verify its originality utilizing the fields of parent ID and DODAG version. In other words, if it was received for the first time, then the receiving node will perform the following steps in sequence. Firstly, it will add the transmitting node (i.e., the node from which it received the DIO message) to its parent list. Secondly, it will compute its rank, which is based on an OF. Lastly, it will update the DIO message, for example, inserting its rank, and accordingly broadcast it to its neighbors. Surprisingly, if this node received the same DIO message but from a different neighbor, then it will compute its new rank and if it is found to be higher than that mentioned earlier (i.e., when being received from the first parent), then it will neither update nor broadcast the DIO message. In contrast, it will update and broadcast the DIO message to its neighbor if its new rank was lower than that of the first parent. Regardless of the above cases, this node will add any node from which it received a DIO message to its parent list. This process will proceed until reaching the leaf node. Except for the leaf nodes, each node transmits periodic DIO messages to its neighbors to keep

4/12

ال RPL متكلم ال Objective function داخله في Metric

RPL — OFO / MHROF

ال LQ بعت لكل ال Neighbor تأخير وبشوف كم وحدة وصلوا وكم وحدة وصلوني
(كم مرة بدي ابعت لحد ما يزبط ال Quality)

$LQ = \frac{1}{P_{ack} * P_{Ack}}$

$ETX = \frac{1}{LQ * NLQ}$ بحيث تكون عالية

* لو كان بيني وبين ال Neighbor أكثر من Wireless devices بالتالي رح اطلع ال Sum اكم كلهم عشان اطلع ال ETX النهائي.

(ETX Routing Metric) فايل في الطاقة وتوايني.

ال ETX بده heavy Process. واحنا هنا low Power devices.
فاستبدلناه بال RSSI.

Figure 8.1 بدي اطلع ال Path
UPward عملية ال Selection والدسم طالعة عندي
downward الاسهم نازلة نزول
كل واحد بيعت DID ال Neighbors الي عندي و الي بوصلو بشيكوا هل موجودين بال table ولا
ال Root بينزل نزول

* ال Rank عندي هو الي بطلعه من ال fuzzy logic

كل DID Message فيها ٣ اشياء Version ID، RPL Instance

اذا وصلني من اكثر من مكان بقرر شو ال Metric لو كان hub count بعرف مين اقل
و بغير ال Parent تاني على اساسه.

الـ DIO بتبلش دايما من الـ Root

بما الـ Upward انه اللي تحت هيه الاولاد واللي فوقهم لما اطلع هيه الـ Parent.

* الـ DAO بنستخدما اذا كان الـ Downward يعني جاي من الـ Leaf

Upward Point to multipoint

Downward Multipoint to point

الـ Downward ما بنستخدما كتير لانه الـ DAO بدها Ack لانه بلشنا من الـ Leaf وراح الـ Root هو الـ Source فعشان اضمن انه بالنهاية وصل للـ Root فلازم Ack.

* اما الـ Upward ما بتحتاج لـ Ack

* لما يدخل عندي Node جديد على الشبكة بده يبعتلي للـ Root والـ Root يرجع DIS Message يحمل DIO عشان يحطله الـ Path

الـ RPL Instance بتكون اما One or more DoDAG.

DoDAG IP ← IPV6 كل واحد Unique لكل واحد

عندي Local Repair انه يكون الحل وديه بيني وبينك وبجاوبك اغير الـ Parent اذا عندي لما يصير Link Fail

Global Repair انه بحتت للـ Root وبديله هو يحلها فبخزن الـ Version number عشان ازبط اذا صار

لما بيعي لاخزن بالـ Neighbor table بخزن DoDAG Version, Parent ID

RPL instance ID DoDAG ID DoDAG version number

* الـ Trickle timer

كل ما يطلع عندي RANK منيح بيقلل الـ Loop Avoidance and detection

خلص هلا الـ Upward والـ Downward جاهز بقدر صرت ابعت البيانات عشان Data Forwarding.

DODAG alive. The trickle timer, which is very important for maintaining the network or keeping the network stable, controls the frequency of DIO messages. It is good to mention that if this frequency/rate goes lower and lower, then the network stability gets better and better. However, when the network is inconsistent, then the intervals of DIO messages are lowered to the bare minimum (Oliveira & Vazao, 2016).

- **Downward Route:** To create downward routes, the RPL uses DAO messages. Applications that require P2P or P2MP communication can use DAO messages as an optional feature. Actually, the RPL allows for two modes of downward traffic, that are, storing and non-storing, with each RPL instance supporting one of these modes. In both modes, and unless the ultimate destination is on the upward route, P2P packets primarily proceed up toward a DODAG root and then down to the final destination. In contrast, the packet will proceed all the way to a DODAG root before going down in the non-storing situation. Before reaching a DODAG root, a shared ancestor (parent) of the source and destination may direct the packet down towards the destination in the storing scenario.
- **Loop Avoidance and Detection:** Within the same DODAG version, the RPL utilizes an efficient loop avoidance algorithm, such as increasing a node's rank, which controls a node's movement through the graph. This is accomplished by utilizing the RPL packet information (header) that is sent along with data packets, which includes the transmitter's rank. A probable loop is indicated by an incompatibility between the routing decision for a packet (upward or downward) and the rank relationship between a pair of nodes. A node initiates a local repair operation when it receives such a packet. For instance, suppose a node gets a packet that is marked as traveling upward. If the transmitter of that packet has a lower rank than the receiving node, then the receiving node can deduce that the packet has not advanced upward and that the DODAG is inconsistent. Therefore, the receiving node initiates a local repair operation.
- **Route Repair:** The RPL uses route repair when inconsistencies emerged, such as loop detection or link failure. Local repair focuses on repairing faults without reconstructing the DODAG from the ground up. Choosing a new parent for a node that has lost connections with its previous parent is an example of such a repair. On the other hand, global repair is a repair mechanism that rebuilds the DODAG from scratch. Actually, increasing or incrementing the DODAG version number indicates that a DODAG root initiates a global repair procedure, thereby initiating a new DODAG version. When local repairs do not produce the best results or optimal solutions, global repair is necessary. The DODAG diverges from its ideal condition each time a local repair is performed. Global repair re-optimizes the topology, but it comes at a price regarding the network performance because the DODAG must be computed again, increasing the control overhead. As the DODAG is rebuilt from scratch, the positions of the nodes in the new DODAG version might be different from those of the old DODAG version. Therefore, nodes' ranks may vary as well.
- **Data Forwarding:** The RPL maintains three types of communications: P2P, P2MP, and MP2P, which is considered as a dominant traffic flow in many LLN applications. This can be done via building and sustaining upward and downward routes.

3. Related Works

In this section, we reviewed the state-of-the-art research carried on designing new OF to enhance RPL performance. As stated earlier, the OF in RPL is responsible for selecting and specifying the preferred parent and/or the best path to reach the DODAG root. However, suggesting an adequate OF in LLNs presents a considerable challenge. Due to the requirements of IoT applications, the performance of LLNs is restricted in terms of the packet delivery ratio, end-to-end delay, efficient energy, and stability of the network. Most of the available literature has investigated the effective or influential elements in routing formation, involving the control message overhead, link quality estimations, and residual energy. It has been demonstrated that the incorporation of the dominant factors can result in a trade-off in the network performance. As a matter of fact, OF may be a single metric such as ETX, HC, energy, or a composite metric. The scrutiny of the prior studies clarified that the OFs that are based on a single metric have inadequacies. Nowadays, the tendency of research on the improvement of RPL OF goes toward combining several metrics based on their impact on the network and application needs.

The authors in (Abreu, Ricardo, & Mendes, 2014) introduced the Energy-Aware Objective Function (EAOF) in which a solution to improve the network's reliability and lifetime of biomedical wireless sensor networks is proposed. In EAOF, smaller nodes are required to work for long time periods without human interference and provide, at the same time, convenient levels for both reliability and Quality of Service (QoS). The proposed solution uses the ETX and residual energy as routing metrics, on each sensor node, to calculate the optimal routes to transmit data packets across the network. Experiment results demonstrated that the suggested approach extends the network lifetime and reduces the energy consumption. Employing fuzzy logic is another design for OF. There is a solution proposed in (Kamgueu, Nataf, & Ndie Djotio, 2015) that utilizes the fuzzy inference system to combine the following routing metrics: ETX, delay, and residual energy of the node, into one single value (quality), to improve QoS and lower energy usage. A real sensor network is deployed over the indoor environment to evaluate the metric. In the proposed approach, the node's rank is obtained by attaining the sum of the preferred parent's rank with respective estimated quality. The obtained results exhibited that the proposed protocol enhanced the network performance in terms of the end-to-end delay, packet loss ratio, routing stability, and energy consumption.

Prolonging the network lifetime and increasing the throughput are the aims of the Sink-to-Sink Coordination Framework (SSCF), which was introduced in (Khan, Lodhi, Rehman, Khan, & Hussain, 2016). Furthermore, in SSCF, the network status seen at a sink is exchanged with its neighboring sinks utilizing the periodic route maintenance messages, issued by RPL. SSCF works through specifying the syntax and semantics of any exchange in the network parameters between sink nodes. To fulfill the throughput optimization, the sink nodes in SSCF set their network size for better load balancing. The simulation findings proved that the new approach improves the throughput, as well as alleviates the energy consumption, therefore, SSCF prolongs the network lifetime. Besides, the packet delivery ratio is increased.

The Objective Function Fuzzy Logic (OF-FL) was designed in (Gaddour, Koubaa, Baccour, & Abid, 2014). The fuzzy system's parameters are used by this OF to generate customized routing decisions based on certain routing metrics. In OF-FL, choosing the best paths to the destination depends on the application requirements. OF-FL handles the fuzzy logic system to utilize the following link and node routing metrics: end-to-end delay, HC, ETX, and the level of battery energy. The preferred parent is defined as a set of IF-THEN rules that take into account all four linguistic variables. The fuzzy rules are estimated so that the association of the neighbor in the fuzzy subsets of neighbors is reverted to the best qualities. The preferred parent is the neighbor with the best quality. The proposed scheme has been proven, by the simulations, to have a significant improvement in the performance in terms of the network lifetime, end-to-end delay, and packet loss ratio. Additionally, OF-FL has been designed with only minimal add-ons, to make it compatible with the native RPL. Another point of strength, in this approach, is that it supports several simultaneous applications with antagonistic requirements. On the other hand, the weaknesses of this approach lie in the fact that OF-FL was not applied in heterogeneous networks, where diverse applications are deployed, besides the increase in the memory usage in return to fuzzy logic implementation.

The authors in (Araújo, et al., 2018) proposed a new design to select routes using a fuzzy logic system, named, Delivery Quality and Context-Aware Objective Functions (DQCA-OF). Four OFs are proposed and particularly selected dynamically based on context information. OFs are generated from the amalgamation of the following routing metrics: ETX, HC, and consumed energy. Moreover, DQCA-OF considers route classifier based on a fuzzy logic system, by approximating the degree of quality of the routes. The route classifier operates as input to the values designated to the routing metrics, guaranteeing consistency in decision-making within the routing process. The simulation results shown that DQCA-OF outperforms other protocols since it provides high reliability, reduces the end-to-end delay for data delivery, and improves the QoS of the network. Besides, it shortens the energy consumption, subsequently, the network lifetime is extended.

A different OF was introduced in (Gozuacik & Oktug, 2015), which is named as Parent-Aware Objective Function (PAOF). It aims at achieving a load-balanced network. PAOF uses both ETX and parents' count (i.e., number of node's candidate parents) as routing metrics to find the optimal route toward the DODAG route. Also, it performs the preferred parent selection and rank's computing. The acquired results indicated that the proposed OF outperforms other OFs by achieving a superior parent load density, end-to-end delay, and parents' change. In

addition, PAOF guarantees the load balancing of the network. Thus, the network lifetime is prolonged. Furthermore, it behaves tolerantly in congestion scenarios.

The closely related OFs to our proposed OF are OF0, MRHOF, and Energy Consumption-aware OF (OF-EC), which were introduced in (Thubert, 2012), (Gnawali & Levis, 2012), and (Lamaazi & Benamar, 2018), respectively. RFC 6552 (Thubert, 2012) introduced the OF0, which is defined as a basic OF that uses the default configurations. It was developed to identify the shortest path possible towards a designated root based on the HC with minimum rank. OF0 recognizes the rank of every candidate neighbor using the DIO message information. The notable challenge that has been drawn against the OF0 is that the nodes nearby the root have higher traffic than other faraway nodes. Therefore, they are liable to consume their energy faster. Consequently, the network lifetime is shortened.

The MRHOF was introduced in RFC 6719 (Gnawali & Levis, 2012), which is more complex than OF0. It chooses the paths with the minimum metric taking advantage of hysteresis to lessen the churn in the light of a change of a metric. Besides, it uses additive metrics to calculate the rank of a node. Thus, it is utilized to determine the route, which has the lowest cost without advertising unnecessary changes of the preferred parent. In general, MRHOF may employ several metrics but the default and most commonly used is the ETX. However, MRHOF can exploit any other routing metrics as stated in (Barthel, Vasseur, Pister, Kim, & Dejean, 2012), such as the latency metric. For the case where the latency metric is considered, the RPL would discover the path with the minimum latency between a node and the DODAG root. In addition, when the DAG Metric Container (DMC), which is a field that exists in the DIO messages and usually configured based on the metric desired, does not contain any metric, then the MRHOF will find a minimum ETX route starting from the node and ending with the root. When the route with minimum rank is found, the MRHOF switches to that route only if it is, by at least a known threshold, shorter than the current route. The ETX starts from one to infinity where the one indicates that the link is more reliable. In fact, as the ETX increases, the link becomes more difficult (i.e., lossier). One of the main downsides of MRHOF is the rapid depletion of nodes' energy induced by the ETX computations required.

In (Lamaazi & Benamar, 2018), a new OF was proposed based on amalgamating metrics utilizing the fuzzy logic, abbreviated as OF-EC. The authors combined the ETX, HC, and EC metrics to select the preferred parent. The new OF is responsible for choosing the best path to the DODAG root. The optimal route has low ETX and EC. The fuzzy logic system of this scheme considers the three aforementioned metrics as input variables. Also, it has one output variable, which is considered the quality of parent. The authors constructed the membership functions for each fuzzy set that belongs to a specific metric. Afterward, a rule evaluation is applied to find the fuzzy output. Interestingly, the OF-EC outperforms the standard RPL OF's performance concerning energy consumption, lifetime of the network, end-to-end delay, messages overhead, packet delivery ratio, and convergence time of the DODAG.

4. The Improved RPL Protocol

This section details the algorithms considered for improving the RPL, which are particularly related to enhancing it's OF. In specific, the assumptions of the improved protocol are given in Section 4.1. Section 4.2 demonstrates the cross-layer algorithm, while Section 4.3 details the appropriate metrics. In Section 4.4, the proposed fuzzy logic system is exemplified comprehensively. The parent selection process is thoroughly explained in Section 4.5. Finally, the limitations and computational burden of our proposed work are discussed in Section 4.6. It is noteworthy to mention that preliminary findings of this research have been presented in an extremely simplified form in (Darabkh, Al-Akhras, & Khalifeh, Oct 2021).

4.1 Assumptions

The following assumptions are borne in mind while designing the FL-HELR-OF:

- IoT network consists of one DODAG root and a large number of routers and leaf nodes that are distributed over a rectangular area.
- All nodes in the network are considered immobile, including the DODAG root.

- The nodes are randomly distributed in an open environment and exchange their data in a radio environment, namely, Unit Disk Graph Medium (UDGM).
- Each DODAG root has a unique DODAG ID and instance ID with no restriction on its energy.
- Each node has a unique IPv6 address, limited power, and constrained memory.
- All nodes have the same initial energy.
- All original RPL configuration parameters' default values are considered in the improved one.

4.2 Cross-layer Design Considered

The concept of cross-layer design relies basically on sharing information or variables between layers. It aims at achieving the efficient use of network parameters besides boosting the performance of the network. Referring to the preceding discussion regarding RPL challenges, it is essential to employ a cross-layer design knowing that the RPL is compatible with IEEE 802.15.4 MAC and IEEE 802.15.4 PHY. The network's QoS is affected by the layers' parameters, which have noticeable influences on the link reliability and energy consumption. Accordingly, choosing low or down layer parameters and then combining them in a composite metric for building a new OF is so advantageous and specifically boosts the selection process of the preferred nodes' parents which will consequently have a positive influence on the network performance. In our proposed work, RSSI and EC are IEEE 802.15.4 PYH layer parameters while the latency metric is an IEEE 802.15.4 MAC layer parameter. The HC is a network layer parameter.

4.3 Metrics of interest

Our novel FL-HELR-OF is developed taking into account improving very affecting dimensions. Firstly, the energy efficiency, where the best route is chosen based on the nodes that have less energy consumption, prolonging the network lifetime. That is done by getting the IEEE 802.15.4 PYH EC parameter involved in our composite metric. Secondly, the reliability, where the best route is said to be reliable if the highest packet delivery ratio is achieved therein. Due to having various environmental conditions, which certainly influence the IoT networks, such as cross-link interferences, large moving objects, weather status, the need for going with the links with the best quality is believed to be in high demand. In FL-HELR-OF, the reliability is considered through getting the important link estimators involved, namely, RSSI. Lastly, real-time delivery, which is an important dimension in IoT networks, especially for applications that require real-time measurements, such as medical and smart city applications. This dimension is considered in our FL-HELR-OF through having the latency metric involved.

In light of the above, the following demonstrates the mathematical aspects related to the metrics used.

- Hop count, which is a network (layer 3), node (topology), and software-based link estimation metric. HC indicates the number of hops required from the sender node until reaching the destination or the DODAG root. Based on that, the preferred parent is the one that has the minimum hop count. The DODAG root has usually zero hop count. On the other hand, the HC of other nodes is found by counting the number of hops between the sender and destination. Referring to the datasheet of Contiki, the range of HC is [1, 7].
- Energy consumption, which is a node-based metric and PHY layer parameter. It represents the energy consumed by nodes due to not only their communications over the network but also their local computations (CPU processing). According to the Contiki motes datasheet, the node's energy range is [0, 250] mJ.

The energy consumed during the communication (i.e., transmission and receiving modes) is calculated as follows (Lamaazi & Benamar, 2018):

$$EC\_comm\,(mJ) = \frac{(T_x \times 20mA) + (T_r \times 22mA)}{RTIMER\_ARCH\_SECOND} \times 3V, \quad (1)$$

where, *Tx* refers to the number of ticks the radio has been in "transmit" mode. *Tr* indicates the number of ticks the radio has been in receive mode (listen). *RTIMER_ARCH_SECOND* represents a Contiki constant, which has a value of 32768 ticks per second.

In addition, the energy consumed during a node's local processing (CPU), which is very important as the fuzzy logic system proposed necessitates several CPU calculations, is computed as (Lamaazi & Benamar, 2018):

$$EC\_CPU\ (mJ) = \frac{(T_{CPU} \times 2mA)}{RTIMER\_ARCH\_SECOND} \times 3V, \quad (2)$$

where, T_{CPU} represents the number of ticks the CPU has been in "active" mode.

In our proposed work, the EC metric is defined as the summation of energies consumed by all nodes along the path to the DODAG root. In other words, the EC metric adds not only a node's energy consumed, as formulated in the aforementioned equations, but also the amount of energy consumed by all nodes located in the path towards the destination (DODAG root).

- Latency, which is a MAC (layer 2) link-based metric. It indicates the assessed time to transmit a packet from the source and receive it by the destination. The latency metric, measured in *ms*, is the summation of all nodes' latencies along the path to the DODAG root (i.e., it is a cumulative metric). It is good to mention that the value of this metric relies primarily on the simulation environment parameters and particularly the network size. In fact, there is nothing available in the Cooja simulator to measure the latency in RPL. It is usually coded and measured based on the reasoning and thinking of the code's implementer or RPL designer. In our work, we code the latency so that it includes the time from the DIO is launched from the sender (DODAG root) until it reaches the destination, which is the most common rationale of RPL designers. Considering the simulation environment, which will be discussed shortly, and after conducting massive simulations, we found out that there is an upper bound for the latency, which is 2 seconds. The latency is calculated as follows:

$$Latency\ (n) = \begin{cases} Latency(P) + Latency_{n \to P}, \\ 0, \quad If\ n\ is\ the\ root \end{cases} \quad (3)$$

where, n and P refer to the node and its parent, respectively. $Latency_{n \to P}$ indicates the latency from a node n to its parent.

- RSSI, which is a hardware-based link estimator and PHY (layer 1) metric. It is adjusted internally by the radio driver after receiving the packet from a particular node. It is used to regulate the volume of radio energy received in a given channel. It is good to state that the RSSI is measured right after receiving the packet. Therefore, it has nothing with the packets' losses. In Contiki, the "*PACKETBUF ATTR RSSI*" variable, from the packet buffer file, is used to read the RSSI value. According to the datasheet of motes in Contiki, obtaining the correct value of RSSI requires adding an offset of -45, which is the front-end gain. Usually, the RSSI's range is between [-100, -50] dBm where higher values of RSSI denote the more reliable links and vice versa. The RSSI is computed as shown in equation (4) (Urama, Fotouhi, & Abdellatif, 2017) bearing in mind that it is calculated upon receiving one packet. When receiving a bundle of packets, the average RSSI should be considered.

$$RSSI_{True\ value} = RSSI_{Last\ packet\ received} + (-45) \quad (4)$$

4.4 Fuzzy Logic Design

Our goal in this work is to propose a novel OF that surpasses other relevant OFs. The combination of many metrics proved its efficiency in selecting the preferred parent and ameliorated the performance of RPL. The fuzzy logic is considered one of the best solutions to combine several metrics. The process of fuzzy logic is to convert multiple input variables into a single output variable. In our case, there are four input metrics, e.g., HC, EC, latency and RSSI. After that, a long fuzzy logic process will be considered, which ends up in having a single output, which is the neighbor's quality. Figure 1 explains the essential steps of the fuzzy logic design of our FL-HELR-OF.

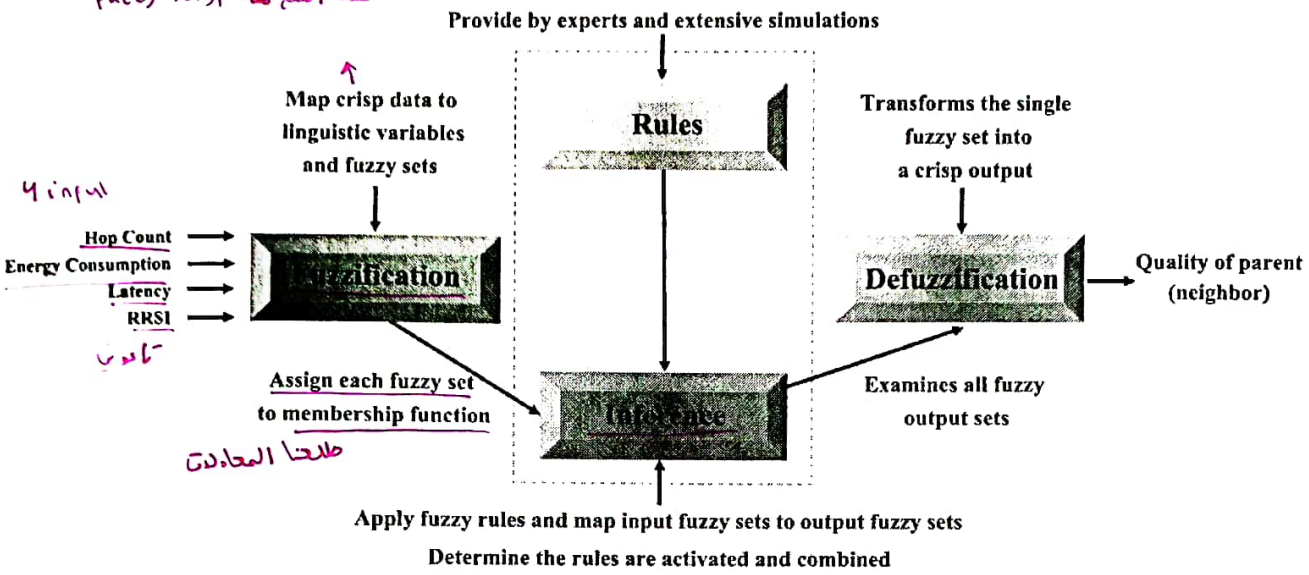


Figure 1: Architecture of fuzzy logic process in FL-HELR-OF

4.4.1. Fuzzification procedure

The process of allocating numerical input variables to fuzzy sets with some degree of membership is known as fuzzification. The fuzzification technique is demonstrated in the following sub-sections.

A. Linguistic variables (fuzzy sets)

The linguistic variable is a variable in which the values are words rather than numbers. Each input metric has three fuzzy sets in FL-HELR-OF, while the output variable has eight fuzzy sets. The following are further details:

i. Fuzzifying the HC input variable. The fuzzy sets of HC input metric have three linguistic names, namely, "*Close*", "*Near*", and "*Far*". Table 2 presents the fuzzy sets along with their ranges, besides the membership function of this metric.

Table 2 : Linguistic variable and fuzzy sets of HC input metric

Fuzzy set	Fuzzy set range	Membership function
Close	0 – 2	R – function
Near	1 - 4	Trapezoidal – function
Far	3 - 7	L - function

ii. Fuzzifying the EC input variable. The three linguistic names for the fuzzy sets of EC input metric are "*Low*," "*Average*," and "*High*." In addition to the membership function of this metric, Table 3 illustrates the fuzzy sets and their ranges.

Table 3: Linguistic variable and fuzzy sets of EC input metric

Fuzzy set	Fuzzy set range (mJ)	Membership function
Low	0 – 100	R – function
Average	50 - 200	Trapezoidal – function
High	150 -250	L - function

iii. Fuzzifying the latency input variable. The fuzzy sets of latency input metric have three linguistic names, which are, "*Short*", "*Average*", and "*Long*". The fuzzy sets, their ranges, including the membership function of this metric, are synopsized in Table 4.

Table 4: Linguistic variable and fuzzy sets of latency input metric

Fuzzy set	Fuzzy set range (ms)	Membership function
Short	0 – 800	R – function
Average	400 - 1600	Trapezoidal – function
Long	1200 - 2000	L - function

iv. Fuzzifying the RSSI input variable. The three linguistic labels for the fuzzy sets of RSSI input metric are "*Connected*," "*Transitioning*," and "*Disconnected*." The fuzzy sets, their ranges, and the membership function of this metric are all shown in Table 5.

Table 5: Linguistic variable and fuzzy sets of RSSI input metric

Fuzzy set	Fuzzy set range (dBm)	Membership function
Connected	(-50) – (-75)	R – function
Transitioning	(-65) - (-95)	Trapezoidal – function
Disconnected	(-85) - (- 100)	L - function

v. Fuzzifying the quality of neighbor output variable. The fuzzy sets of the quality of neighbor output comprise eight linguistic names, which are, "*Awful*", "*Low Bad*", "*Bad*", "*Reasonable*", "*Average*", "*Good*", "*Very Good*", and "*Excellent*". Table 6 shows the fuzzy sets, their ranges, together with the membership function of this metric.

Table 6: Linguistic variable and fuzzy sets of Quality output metric

Fuzzy set	Fuzzy set range	Membership function
Awful	0 – 20	Triangle – function
Low Bad	10 - 30	Triangle – function
Bad	20 – 40	Triangle - function
Reasonable	30 -50	Triangle - function
Average	40 – 60	Triangle - function
Good	50 - 70	Triangle - function
Very Good	60 – 80	Triangle - function
Excellent	70 - 100	Triangle - function

B. Membership functions

The membership function encourages graphic representation of the fuzzy set. The degree of membership falls between 0 and 1. A variety of factors influence the choice of membership function, including massive and lengthy simulations that rely on trial-and-error procedures, prior literature, application requirements, as well as device datasheets. The Trapezoidal function is chosen for the input metrics of our design since it is broadly implemented in fuzzy logic (Brule, 1985). Furthermore, there are two special functions used, which are derived from the Trapezoidal function, namely, R-functions and L-functions. In addition, the Triangle-function is another commonly utilized function. In our design, it is applied to the variable's output. Figures (2 – 5) point out the membership functions of the four-input metrics whereas Figure 6 displays the membership function of the quality output variable. The equations (5– 16) exhibits thorough membership function computations for the four-input metrics.

Tip: The definitions of the most commonly used Fuzzy logic membership functions are as follows:

$$\text{Trapezoidal}(x; a, b, c, d) = \begin{cases} 0, & x < a \text{ or } x > d \\ \left(\frac{x-a}{b-a}\right), & a \le x \le b \\ 1, & b \le x \le c \\ \left(\frac{d-x}{d-c}\right), & c \le x \le d \end{cases} \quad (1)$$

$$\text{R-Function}(x; c, d) = \begin{cases} 0, & x > d \\ \left(\frac{d-x}{d-c}\right), & c \le x \le d \\ 1, & x < c \end{cases} \quad (2)$$

$$\text{L-Function}(x; c, d) = \begin{cases} 0, & x < a \\ \left(\frac{x-a}{b-a}\right), & a \le x \le b \\ 1, & x > b \end{cases} \quad (3)$$

$$\text{Triangular}(x; a, b, c) = \begin{cases} \left(\frac{x-a}{b-a}\right), & a \le x \le b \\ \left(\frac{c-x}{c-b}\right), & b \le x \le c \\ 0, & otherwise \end{cases} \quad (4)$$

- **The membership function of HC input metric:**

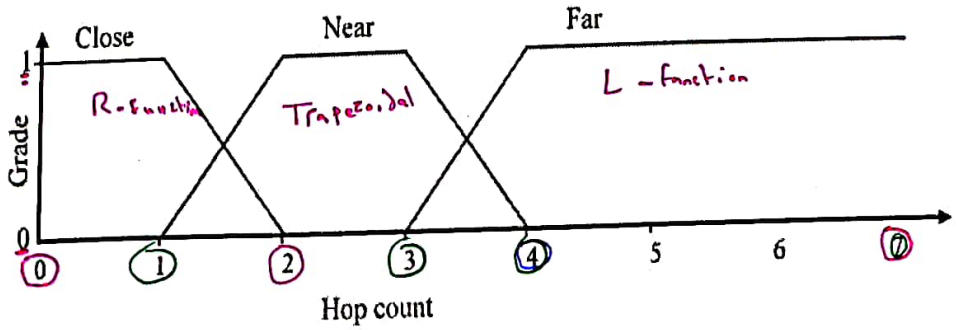


Figure 2: Membership functions of the HC input metric

$$Close(HC) = \begin{cases} 1, & HC < 1, \\ \dfrac{2-HC}{2-1}, & 1 \le HC \le 2, \\ 0, & HC > 2. \end{cases} \quad (5)$$

$$Near(HC) = \begin{cases} 0, & HC < 1, \\ \dfrac{HC-1}{2-1}, & 1 \le HC \le 2, \\ 1, & 2 \le HC \le 3, \\ \dfrac{4-HC}{4-3}, & 3 \le HC \le 4, \\ 0, & HC > 4. \end{cases} \quad (6)$$

$$Far(HC) = \begin{cases} 0, & HC < 4, \\ \dfrac{HC-3}{4-3}, & 3 \le HC \le 4, \\ 1, & HC > 4. \end{cases} \quad (7)$$

- **The membership function of EC input metric:**

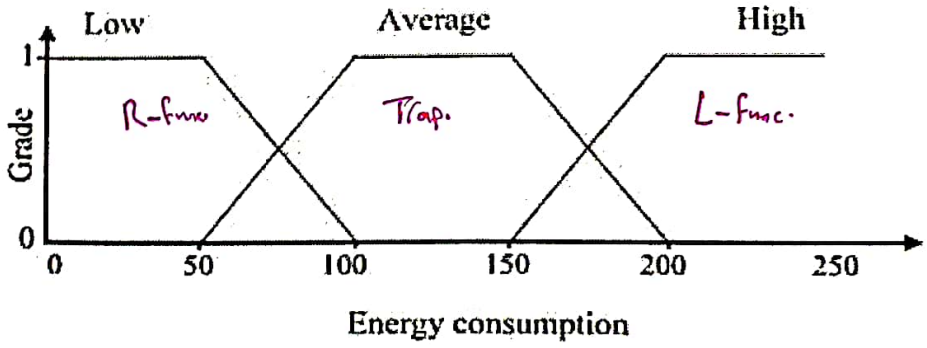


Figure 3: Membership functions of the EC input metric

$$Low\ (EC) = \begin{cases} 1, & EC < 50, \\ \dfrac{100 - EC}{100 - 50}, & 50 \le EC \le 100, \\ 0, & EC > 100. \end{cases} \quad (8)$$

$$Average\ (EC) = \begin{cases} 0, & EC < 50, \\ \dfrac{EC - 50}{100 - 50}, & 50 \le EC \le 100, \\ 1, & 100 \le EC \le 150, \\ \dfrac{200 - EC}{200 - 150}, & 150 \le EC \le 200, \\ 0, & EC > 200. \end{cases} \quad (9)$$

$$High(EC) = \begin{cases} 0, & EC < 150, \\ \dfrac{EC - 150}{200 - 150}, & 150 \le EC \le 200, \\ 1, & EC > 200. \end{cases} \quad (10)$$

- **The membership function of latency input metric:**

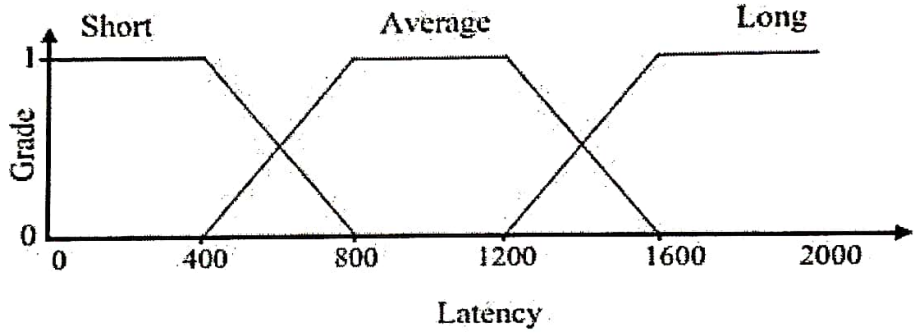


Figure 4: Membership functions of the Latency input metric

$$Short\ (latency) = \begin{cases} 1, & latency < 400, \\ \dfrac{800 - latency}{800 - 400}, & 400 \le latency \le 800, \\ 0, & latency > 800. \end{cases} \quad (11$$

$$Average\ (latency) = \begin{cases} 0, & latency < 400, \\ \dfrac{latency - 400}{800 - 400}, & 400 \le latency \le 800, \\ 1, & 800 \le latency \le 1200, \\ \dfrac{1600 - latency}{1600 - 1200}, & 1200 \le latency \le 1600, \\ 0, & latency > 1600. \end{cases} \quad (12$$

$$Long(latency) = \begin{cases} 0, & latency < 1200, \\ \dfrac{latency - 1200}{1600 - 1200}, & 1200 \le latency \le 1600, \\ 1, & latency > 1600. \end{cases} \quad (13)$$

- **The membership function of RSSI input metric:**

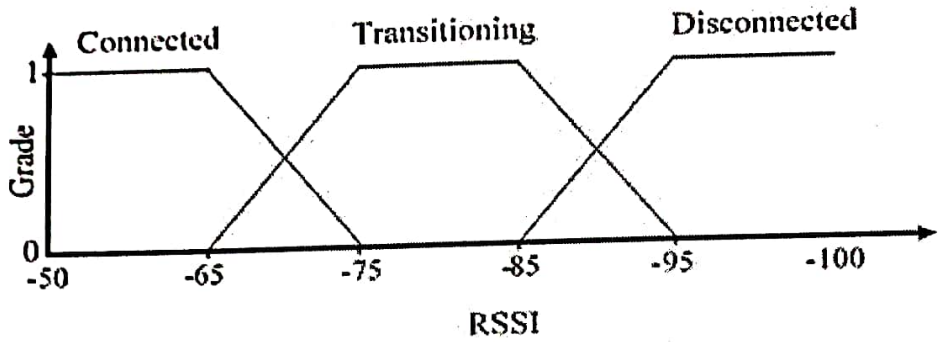


Figure 5: Membership functions of the RSSI input metric

$$Connected\ (RSSI) = \begin{cases} 1, & RSSI > -65, \\ \dfrac{RSSI - (-75)}{(-65) - (-75)}, & -75 \le RSSI \le -65, \\ 0, & RSSI < -75. \end{cases} \quad (14)$$

$$Transitioning\ (RSSI) = \begin{cases} 0, & RSSI > -65, \\ \dfrac{RSSI - (-65)}{(-75) - (-65)}, & -75 \le RSSI \le -65, \\ 1, & -85 \le RSSI \le -75, \\ \dfrac{RSSI - (-95)}{(-85) - (-95)}, & -95 \le RSSI \le -85, \\ 0, & RSSI < -95. \end{cases} \quad (15)$$

$$Disconnected\ (\text{RSSI}) = \begin{cases} 0, & RSSI > -85, \\ \dfrac{RSSI - (-85)}{(-95) - (-85)}, & -95 \le RSSI \le -85, \\ 1, & RSSI < -95. \end{cases} \quad (16)$$

- **The membership function of the quality of neighbor output metric:**

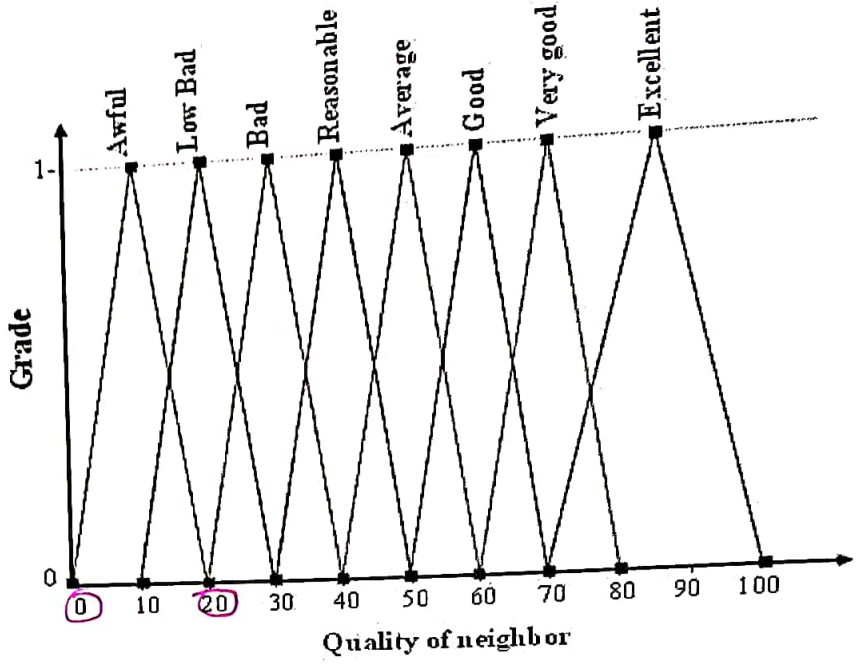


Figure 6: Membership functions of the quality of neighbor output metric

4.4.2. Fuzzy inference system

A. Rule evaluation

In this stage of fuzzy logic, IF-THEN rules are constructed. The number of generated rules in FL-HELR-OF is set based on the number of input metrics and the number of fuzzy sets associated with each metric. Hence, the number of generated rules equals 81 (3^4) rules. Table 7 displays a sample of these rules where appendix A details the complete fuzzy rule base considered in FL-HELR-OF. The fuzzy rule base, shown in Table 7, consists of four fuzzy sets of each metric whereas the last column is the fuzzy set of the output variable that varies from "*Awful*" to "*Excellent*". It is quite interesting to mention that these rules are constructed based on human expertise, experience simulation evaluation, and application requirements.

Table 7: Fuzzy rule base

Rule No.	Input metrics				Quality Output
	HC	EC	Latency	RSSI	
1	Close	Low	Short	Connected	Excellent
2	Near	Low	Short	Connected	Very Good
37	Close	Average	Average	Connected	Good
32	Near	Average	Short	Transitioning	Average
26	Near	Low	Long	Disconnected	Reasonable
44	Near	Average	Average	Disconnected	Bad
78	Far	High	Long	Transitioning	Low Bad
81	Far	High	Long	Disconnected	Awful

B. Aggregation

The Mamdani inference system (Mamdani & Assilian, 1975) is used in our research. In specific, we appl a fuzzy aggregation operator to merge the same fuzzy sets output of each rule into a single output fuzzy set bearin

in mind that the "*AND*" and "*OR*" logical operators denote for "*Minimum*" and "*Maximum*", respectively. For example, the output fuzzy set "*Average*" combined many rules. In our design, we suppose that five rules are fired, thereby aggregating them using the "*Maximum*" operator. The following formula shows how to measure the "*Average*" fuzzy set output considering that five samples out of eleven are fired for the "*Average*" output quality:

$$Average = Maximum \begin{pmatrix} Minimum(Far\_HC, Low\_EC, Long\_latency, Connected\_RSSI), \\ Minimum(Near\_HC, High\_EC, Short\_latency, Connceted\_RSSI), \\ Minimum(Close\_HC, High\_EC, Long\_latency, Connected\_RSSI), \\ Minimum(Near\_HC, Average\_EC, Average\_latency, Transitioning\_RSSI), \\ Minimum(Near\_HC, Average\_EC, Short\_latency, Transitioning\_RSSI). \end{pmatrix} \quad (17)$$

4.4.3. Defuzzification process

In the final step of the fuzzy logic, the FL-HELR-OF is expected to convert the aggregated fuzzy output obtained from section 4.4.2 into a single crisp value. This step assigns the quality of the neighbor (parent), whose value ranges from 0 to 100. The center of gravity algorithm is the one adopted for the defuzzification process of our FL-HELR-OF, which finds the center of the aggregated rule (fuzzy set output). This crisp output yields the quality of each node's neighbor (parent) which ultimately helps in obtaining the preferred parent along the path to the DODAG root. It is found as follows (Landi, 2002):

$$Neighbor\ Quality = \frac{\sum_{i=1}^{N} W_i \times \mu A(W_i)}{\sum_{i=1}^{N} \mu A(W_i)}, \quad (18)$$

where, N represents the number of fired rules, W_i indicates the domain value related to the i^{th} rule (the center of gravity for each output's fuzzy set), and $\mu A(W_i)$ shows the predicate truth in the domain.

In what follows, the procedure of computing the quality of the node using fuzzy logic is illustrated. For instance, let us assume that the N_y is the candidate parent, of the node N_x, which has the following values, HC =1, EC = 60 mJ, latency = 500 ms, and RSSI = -70 dBm. As per the FL-HELR-OF membership functions, shown in Figures (3-6), the HC metric fuzzy set is, "*Close*", therefore, when referring to equation (5), the result of the membership function is 1. Moreover, the EC metric has two fuzzy sets ("*Low*" and "*Average*") and after substituting its value (60 mJ) into equations (8) and (9), the results of the membership functions are 0.8 and 0.2, respectively. The latency metric has also two fuzzy sets ("*Short*" and "*Average*") taking into account that the results of the membership functions are 0.75 and 0.25, respectively, after, of course, substituting 500 ms into equations (11) and (12). The last metric, which is RSSI, has two fuzzy sets ("*Connected*" and "*Transitioning*"), therefore, the results of the membership functions are 0.5 and 0.5 respectively, after substituting -70 dBm into equations (14) and (15). Referring to the fuzzy rule base, shown in Appendix A, we have eight fuzzy rules for the output quality, that are, rules 1, 4, 10, 13, 28, 31, 37, and 40 which matched perfectly with the inputs' fuzzy sets combinations. These rules activate the "*Excellent*", "*Very Good*", "*Good*", and "*Average*" output fuzzy sets. By applying the aggregation, detailed in subsection 4.4.2 (B), we found that the fuzzy rules of the output quality, namely, rules 1, 4, 10, 13, 28, 31, 37, and 40 have the values of 0.5, 0.5, 0.25, 0.25, 0.2, 0.2, 0.2, and 0.2 which belong to "*Excellent*", "*Very Good*", "*Very Good*", "*Good*", "*Very Good*", "*Good*", "*Good*", and "*Average*" output quality fuzzy sets, respectively. In the penultimate step, particularly applying the exact aggregation, shown in equation (17), the aggregated values of the "*Excellent*", "*Very Good*", "*Good*", and "*Average*" distinctive output quality fuzzy sets are 0.5, 0.5, 0.25, and 0.2, respectively. Considering the center of gravity algorithm for the results of the output quality membership functions and substituting the aforementioned values into equation (18), the defuzzified crisp output (Q) is found precisely as follows:

$$Q = \frac{0.5\times 85 + 0.5\times 70 + 0.25\times 60 + 0.2\times 50}{0.5+0.5+0.25+0.2} = 70.69$$

After the quality of each node n's neighbor (parent) is found, the rank of this node $(Rank_{node_n})$ can be found, which is parameterized by its parent rank $(Rank_{parenhod_n})$ and the *rank-increase* as follows (Winter, et al., 2012):

$$Rank_{nod_n} = Rank_{parenhod_n} + Rank-increase, \quad (19)$$

where the $Rank-increase$ is parametrized further by *step* (i.e., the quality of node n's parent) and $Min-Hop-Rank-Increase$, a Contiki parameter (specifically a field of DODAG configuration option in the DIO message), which has a default value of 256. In fact, there is another relevant Contiki parameter called $Max-Hop-Rank-Increase$, which has a value of 1792 (7 times the former one), confirming what is mentioned earlier (i.e., the HC ranges from 1 to 7). However, according to Contiki computations, the rank of the DODAG root $(Rank_{root})$ is assumed to be 256, which results in having the $Rank_{node_n}$ always exceeded 256.

Consequently, the true integer rank of a node n (TIR_{node_n}) is found as (Winter, et al., 2012):

$$TIR_{node_n} = \left\lfloor \frac{Rank_{node_n}}{Min-Hop-Rank-Increase} \right\rfloor. \quad (20$$

A comprehensive example that details how the FL-HELR-OF works is provided in Appendix B.

4.5 Parent Selection Process

To discover the desired path from each node to the DODAG root in RPL and for data transmission purposes a graph (DODAG) is formed. The root node, which is usually the data gathering node, is where the DODAG structure is built (the Sink). Using the DIO message, the root initiates the distribution of structure-related information. Nodes, located within the root's communication range, will receive and process the DIO message and consequently decide whether (or not) to join the structure based on the characteristics of the DODAG. In fact, the RPL instance is made up of one or more DODAGs with a specific RPL instance ID. The DODAG root is the one in charge of setting the RPL instance ID, which is an 8-bit value that specifies which RPL instance the DODAG belongs to. A RPL node can only be a member of one DODAG at a time in a RPL instance. Actually, each RPL instance runs independently of other RPL instances and hence uses the same OF (i.e., the OF is unique in a certain RPL instance). A DODAG ID identifies a DODAG root and specifically is a unique identifier for each RPL instance. A DODAG is uniquely identified by a 128-bit IPv6 address assigned by a DODAG root.

The entry, which identifies a DODAG, is crucial and has the pair of RPL instance ID and DODAG ID. A DODAG version number, on the other hand, is an 8-bit unsigned integer regulated by the DODAG root. The DODAG version number is incremented by the root to establish a new version of the DODAG. To put it another way, incrementing the DODAG version number signals that the DODAG root commences a global repair procedure resulting in a new DODAG version. It is worth mentioning that when inconsistencies arise, such as loop detection or link failure, route repairs are carried out. Actually, local repair focuses on fixing flaws rather than completely rebuilding the DODAG. When local repairs do not yield the best outcomes or provide the finest answers, a global repair is required. Based on the above, a DODAG version is a single DODAG iteration with a certain DODAG ID. In essence, the entry, which consists of the attributes, namely, RPL instance ID, DODAG ID, and DODAG version number, distinguishes a DODAG version.

To this end, the DODAG root sends out a DIO message to its neighbors, who then use the fields of parent ID and DODAG version to verify its authenticity. In other words, the neighbor (i.e., receiving node) will verify whether (or not) the first time it has been received. In particular, the neighbor node will go through the procedure in order. To begin, it will add the transmitting node (i.e., the node from which it got the DIO message) to its parent list. Second, through its CPU processing, it will run the fuzzy logic procedure proposed and specifically apply equation (18) to compute the quality of this parent. Third, it will use equation (19) to get its rank. Finally, it will make proper changes to the DIO message, such as inserting its rank, and consequently broadcast it to its neighbor

If this node receives the same DIO message from a different neighbor, then it will repeat the aforementioned calculations to determine its new rank (i.e., using equations (18) and (19)). It will not update or broadcast the DIO message if its new rank is discovered to be higher than its previous one. If its new rank is lower than its previous one, however, it will update and send the DIO message to its neighbor. This node will add any node from which it got a DIO message to its parent list, regardless of whether any of the aforementioned scenarios occur. This process will continue until the leaf node is reached. Each node sends DIO messages to its neighbors regularly to keep the DODAG alive except the leaf nodes. The trickle timer regulates the frequency of DIO messages for the reason of sustaining or stabilizing the network. In different words, when the frequency/rate of DIO messages drops, the network stability improves. When the network becomes severely unstable, however, the intervals between DIO messages are reduced to the absolute minimum (i.e., the DIO frequency is set to the bare maximum). Figure 7 depicts the major steps of the parent selection process used in our study.

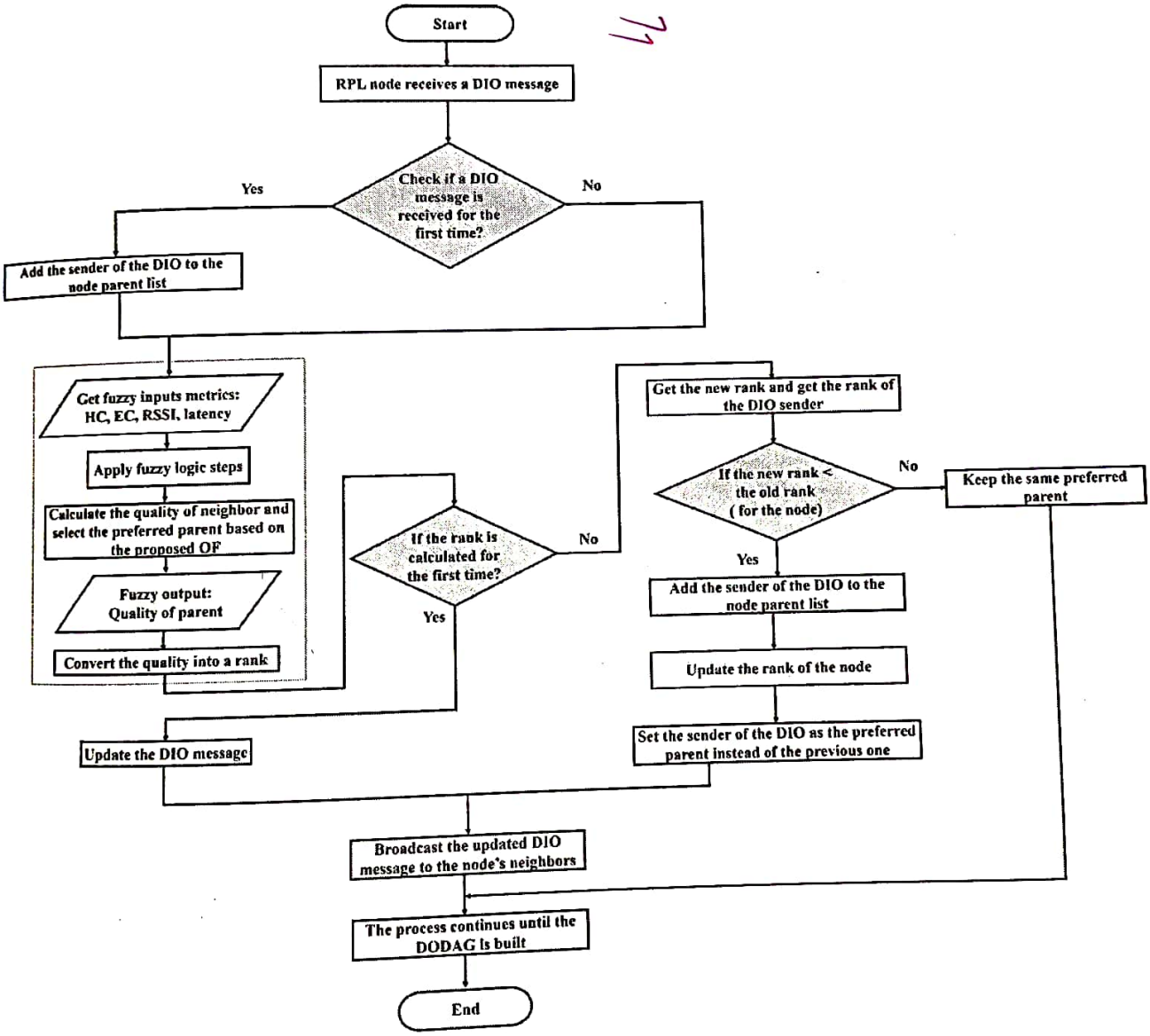


Figure 7: The process of node's parent selection

Algorithm 1 illustrates the process of implementing the FL-HELR-OF in a new Contiki OS file called "fuzzy-of.c". Applying the fuzzy logic dessin, discussed earlier, requires receiving information of the four-input metrics through the DIO message. Fortunately, these metrics will be available in DMC, which is an option in the DIO message. Thereafter, the receiving node will update the receiving metrics based on its effect. For instance, it will increment the HC by one. As previously mentioned, a node will update the DIO before broadcasting it. It is good to emphasize on the point that its effect on the aforementioned metrics will certainly be part of the DIO update (i.e., its new metrics will be inserted in the DMC option) bearing in mind that some metrics are updated after executing the fuzzy logic design proposed. However, the next step involves applying the fuzzy logic design which ends up in finding the parent's quality. Algorithm 2 explains extensively how the quality of the parent is evaluated (i.e., the step of the rank-increase, shown in equation (18). Excitingly, there is a field called OCP, which is a 16-bit unsigned integer and required for identifying the OF, which is found in the DODAG configuration option of the DIO message. In other words, through OCP, we can decide which OF can be used. For instance, some OCP common values define different OFs which are developed based on single and composite metrics. In our work, we set a new value indicating the employment of a new OF which is based on our novel fuzzy logic system. Applying the fuzzification stage of our fuzzy logic design results in having the membership function for each input. Then, we apply Mamdani's fuzzy inference for rule evaluation. Accordingly, the defuzzification process gets started. Finally, the crisp output is assessed, which represents the quality of the neighbor (parent). After that, the rank of the node is computed as detailed in equations (19) and (20). Algorithm 1 includes further the scenario where a node receives a DIO message from more than a parent. In this case, the process of choosing the best one (i.e., the one that has the lowest rank) is abundantly explained. In addition, updating the node's parent list and the DIO message is clearly discussed.

4.6 Limitations and Computational Burden

Unfortunately, the majority of currently proposed RPL protocols, including ours, are computationally heavy and sophisticated, necessitating more storage and processing capabilities. In fact, this is due to the employment of optimization methods, a dramatic increase of DIO message length, and excessive use of control messages particularly, at mobility usage, and, finally, the implementation overheads. Besides, the Cooja simulator motes have limited flash memory. For instance, Sky mote has a flash memory of 48 KB ROM, Z1 mote has a flash memory of 96 KB ROM, and Wis mote has a flash memory of 256 KB ROM, which make the researchers pay attention to the computational complexity influenced by their suggested ideas. It should not be overlooked that the RPL is fully implemented in the Cooja simulator, with some attempts now underway to fully integrate it in Ns3. As far as the optimization method, used in this work, is concerned, we propose a novel fuzzy logic system that basically considers four input metrics, which is surely a big challenge. In particular, for each input metric, we examine three sets of values. Thus, the fuzzy logic system has 81 rules (3^4). Additionally, the best interval for each set in the membership is chosen based on trial-and-error procedures, requiring a tremendous number of simulations. Actually, evaluating some network performance metrics such as the energy consumption needs at least 24 hours for a run to end. Therefore, it took quite a bit of time to test the validity and effectiveness of our proposed protocol taking into consideration the elimination of the possibility of any erroneous individual results through conducting a colossal number of simulation experiments.

5. Simulation Results and Discussion

In this section, the performance of the FL-HELR-OF is thoroughly assessed by conducting an immense number of experiment simulations detailed as follows. First of all, the simulation environment, including setup and parameters, is illustrated. Thereafter, the network performance metrics are defined. Following that, the simulation results are extensively discussed.

5.1 Simulation Environment Setup and Parameters

The simulations are carried out using Cooja simulator, which is operated under Contiki OS, on a laptop with Intel Core i5 processer working at 1.9 GHz with 4GB RAM and runs Ubuntu 12.1(32 bits). In our simulation experiments, we assume that the network consists of 100 nodes and one DODAG root. These nodes are randomly distributed over a 600 m×600 m area. Each RPL router transmits one packet to the DODAG root every 60 seconds during the simulation period. We have undertaken the simulation several times to eliminate the validity of any anomalous individual result. UDGM is used, which is a Cooja feature that adds looseness to the wireless media, allowing for more realistic results. We set the transmission and interference ranges for the nodes to 50 meters and 100 meters, respectively. The motes that are employed in the simulation are Zolertia 1 (Z1), which has a microcontroller MSP 430 with 2.4 GHz wireless transceiver Chipcon CC2420 and has 96k of flash memory and 16-bit RISC architecture upgraded to 20-bit to increase the memory size. Each simulation run took 1 hour long, except the one associated with the energy consumption results which took 24 hours. All other parameters are kept as the defaults of the standard. Table 8 outlined simulation parameters.

5.2 Performance Metrics

The performance metrics used in evaluating the performance of our improved RPL protocol are as follows:

- *Packet delivery ratio,* which represents the ratio of total packets received at the DODAG root to total packets sent from the nodes. The metric's primary purpose is to provide feedback on the network's reliability. In other words, impairments might be known, whether there are any malfunctioning devices or inappropriate connections over the network that may affect the network connectivity and cause downtime or performance issues, thereby estimating the network reliability.
- *Control message overhead,* which represents the total number of control messages (DIS, DIO, and DAO), which are generated by DODAG participating nodes. This metric is influential and particularly indicates the stability and consistency of the network.
- *Latency,* which identifies the average end-to-end delay imposed at each packet during its way from a node to the DODAG root. It is chosen as a network performance since a long delay (waiting) for the acknowledgments of reliable packets (like DAO) from the destination causes many unnecessary retransmissions. Consequently, the packet delivery ratio and the throughput are severely affected, resulting in degrading the network performance.
- *Energy consumption,* which refers to the total energy consumed by a node during not only its communication process over the network, but also its local processing (CPU). As LLNs compose of low-powered devices, this metric is very crucial due to its associativity with the network lifetime, which refers to how long nodes can stay alive without replacing or recharging their batteries.
- *Average hop count,* which represents the average number of hops between each node and its DODAG root. This metric is of interest as the lower number of hops during the packet transmissions, the better network performance.

5.3 Results and Discussion

This section presents and discusses the simulation results. We use "perl" files to evaluate the raw data obtained from the Cooja simulator (Cooja test logs). We investigate the behavior of OF0, MRHOF, OF-EC, and our proposed OF by varying the number of sensor nodes deployed. In specific, we deploy, over a network, up to 100 nodes to verify the scope of efficiency of the aforementioned OFs (i.e., considering light and heavy loaded network environment). Figure 8 shows the shape of the network model after having 25 and 100 nodes deployed. However, in what follows, we will illustrate five network scenarios that are related to the aforementioned performance metrics, in sequence.

and different network densities.

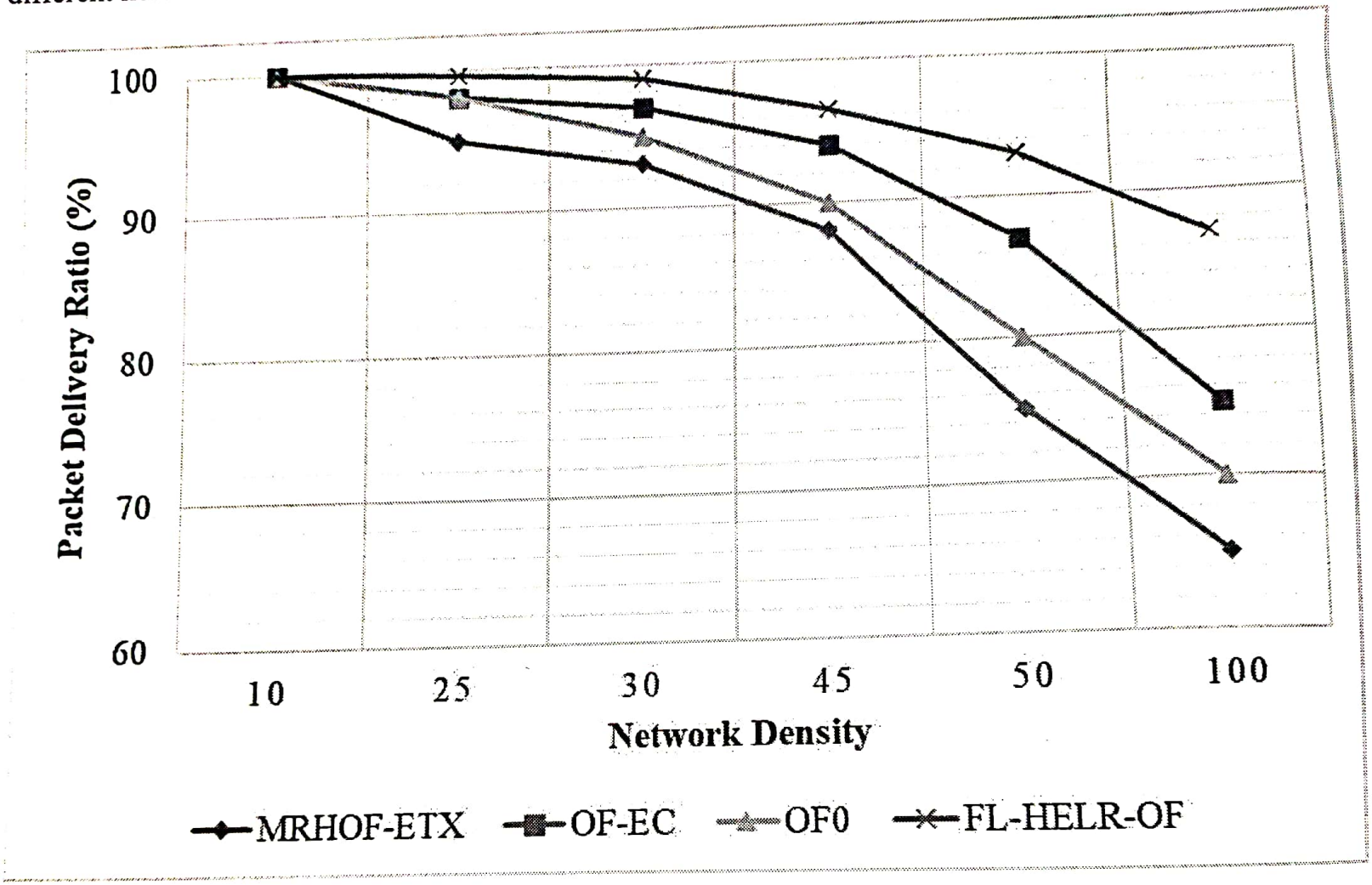


Figure 9: Packet delivery ratio against various network densities and OFs

* كل ما زدنا ← بار wireless بزيد التداخل بينهم فبقل ار Packet Delivery .

Appendix A. Fuzzy Rule Base

The number of created rules in FL-HELR-OF is determined by the number of input metrics and fuzzy sets associated with each metric, as previously noted. As a result, there are 81 rules produced in total. The fuzzy rule base considered in FL-HELR-OF is detailed in Table A.1.

Table A.1 Fuzzy rule base

Rule No.	HC	EC	Latency	RSSI	Output Quality
1	Close	Low	Short	Connected	Excellent
2	Near	Low	Short	Connected	Very Good
3	Far	Low	Short	Connected	Very Good
4	Close	Low	Short	Transitioning	Very Good
5	Near	Low	Short	Transitioning	Good
6	Far	Low	Short	Transitioning	Good
7	Close	Low	Short	Disconnected	Good
8	Near	Low	Short	Disconnected	Good
9	Far	Low	Short	Disconnected	Average
10	Close	Low	Average	Connected	Very Good
11	Near	Low	Average	Connected	Good
12	Far	Low	Average	Connected	Good
13	Close	Low	Average	Transitioning	Good
14	Near	Low	Average	Transitioning	Average
15	Far	Low	Average	Transitioning	Reasonable
16	Close	Low	Average	Disconnected	Good
17	Near	Low	Average	Disconnected	Reasonable
18	Far	Low	Average	Disconnected	Reasonable
19	Close	Low	Long	Connected	Very Good
20	Near	Low	Long	Connected	Good
21	Far	Low	Long	Connected	Average
22	Close	Low	Long	Transitioning	Good
23	Near	Low	Long	Transitioning	Reasonable
24	Far	Low	Long	Transitioning	Reasonable
25	Close	Low	Long	Disconnected	Average
26	Near	Low	Long	Disconnected	Reasonable
27	Far	Low	Long	Disconnected	Bad
28	Close	Average	Short	Connected	Very Good
29	Near	Average	Short	Connected	Good
30	Far	Average	Short	Connected	Good
31	Close	Average	Short	Transitioning	Good
32	Near	Average	Short	Transitioning	Average
33	Far	Average	Short	Transitioning	Reasonable
34	Close	Average	Short	Disconnected	Good
35	Near	Average	Short	Disconnected	Reasonable
36	Far	Average	Short	Disconnected	Reasonable
37	Close	Average	Average	Connected	Good
38	Near	Average	Average	Connected	Average
39	Far	Average	Average	Connected	Reasonable
40	Close	Average	Average	Transitioning	Average
41	Near	Average	Average	Transitioning	Average
42	Far	Average	Average	Transitioning	Bad
43	Close	Average	Average	Disconnected	Reasonable

44	Near	Average	Average	Disconnected	Bad
45	Far	Average	Average	Disconnected	Low Bad
46	Close	Average	Long	Connected	Good
47	Near	Average	Long	Connected	Reasonable
48	Far	Average	Long	Connected	Reasonable
49	Close	Average	Long	Transitioning	Reasonable
50	Near	Average	Long	Transitioning	Bad
51	Far	Average	Long	Transitioning	Low Bad
52	Close	Average	Long	Disconnected	Reasonable
53	Near	Average	Long	Disconnected	Low Bad
54	Far	Average	Long	Disconnected	Low Bad
55	Close	High	Short	Connected	Very Good
56	Near	High	Short	Connected	Good
57	Far	High	Short	Connected	Average
58	Close	High	Short	Transitioning	Good
59	Near	High	Short	Transitioning	Reasonable
60	Far	High	Short	Transitioning	Reasonable
61	Close	High	Short	Disconnected	Average
62	Near	High	Short	Disconnected	Reasonable
63	Far	High	Short	Disconnected	Good
64	Close	High	Average	Connected	Good
65	Near	High	Average	Connected	Reasonable
66	Far	High	Average	Connected	Reasonable
67	Close	High	Average	Transitioning	Reasonable
68	Near	High	Average	Transitioning	Bad
69	Far	High	Average	Transitioning	Low Bad
70	Close	High	Average	Disconnected	Reasonable
71	Near	High	Average	Disconnected	Low Bad
72	Far	High	Average	Disconnected	Low Bad
73	Close	High	Long	Connected	Average
74	Near	High	Long	Connected	Reasonable
75	Far	High	Long	Connected	Bad
76	Close	High	Long	Transitioning	Reasonable
77	Near	High	Long	Transitioning	Low Bad
78	Far	High	Long	Transitioning	Low Bad
79	Close	High	Long	Disconnected	Bad
80	Near	High	Long	Disconnected	Low Bad
81	Far	High	Long	Disconnected	Awful

Appendix B. Comprehensive Example of FL-HELR-OF

To illustrate the operation of our modified RPL protocol, we assume having a DODAG which consists of a root node along with many nodes, participate in the building process, with the target of establishing the best routes between these nodes and that DODAG root. In particular, Figure B.1 (a) shows the distribution of the nodes before the DODAG is constructed while Figure B.1 (b) shows the routes established between all nodes and the DODAG root utilizing FL-HELR-OF. Not only to this extent, but rather Table B.1 provides complete steps that clearly show how this DODAG is built detailing that the first column represents participating nodes, which receive the DIO messages in sequence. The tuples are listed based on the order of receiving the DIO messages at participating nodes. For instance, the last two tuples refer to the case where the DIO message is received at the leaf nodes (i.e., nodes H and M). The second column shows the originator of the DIO messages. The possible communications between receiving nodes and their neighbor (i.e., DIO message originator) are presented in the third column. The next eight columns refer to a neighbor (i.e., DIO message originator) information. In particular, the fourth, fifth, sixth, and seventh columns indicate the measured metrics HC, EC, latency, and RSSI, respectively, where the values of the latest three metrics are chosen by authors. The next column refers to the quality of neighbor, which is computed utilizing equation (18). The next column represents whether the DIO message is received at this node for the first time or not. The next column belongs to the old parent along with its rank. The *rank-increase* is found in the eleventh column, which is the summation of the neighbor quality and *Min-Hop-Rank-Increase* (256). The next column shows the parent rank, which is extracted from the received DIO message. To this extent, the next two columns represent the rank of the DIO message receiving node and its true integer rank as illustrated in equations (19) and (20), respectively. The next column represents the status of the DIO message at the receiving node. In other words, will this receiving node discard this DIO message, due to receiving it from a parent that has a higher rank than its current parent, or update and then broadcast it as of being received either for the first time or from a parent that has a lower rank than its current one? The last column represents the best parent of the DIO message receiving node. The tenth and last columns will be very useful to distinguish between the cases where a receiving node keeps its old parent or switches totally to a new one.

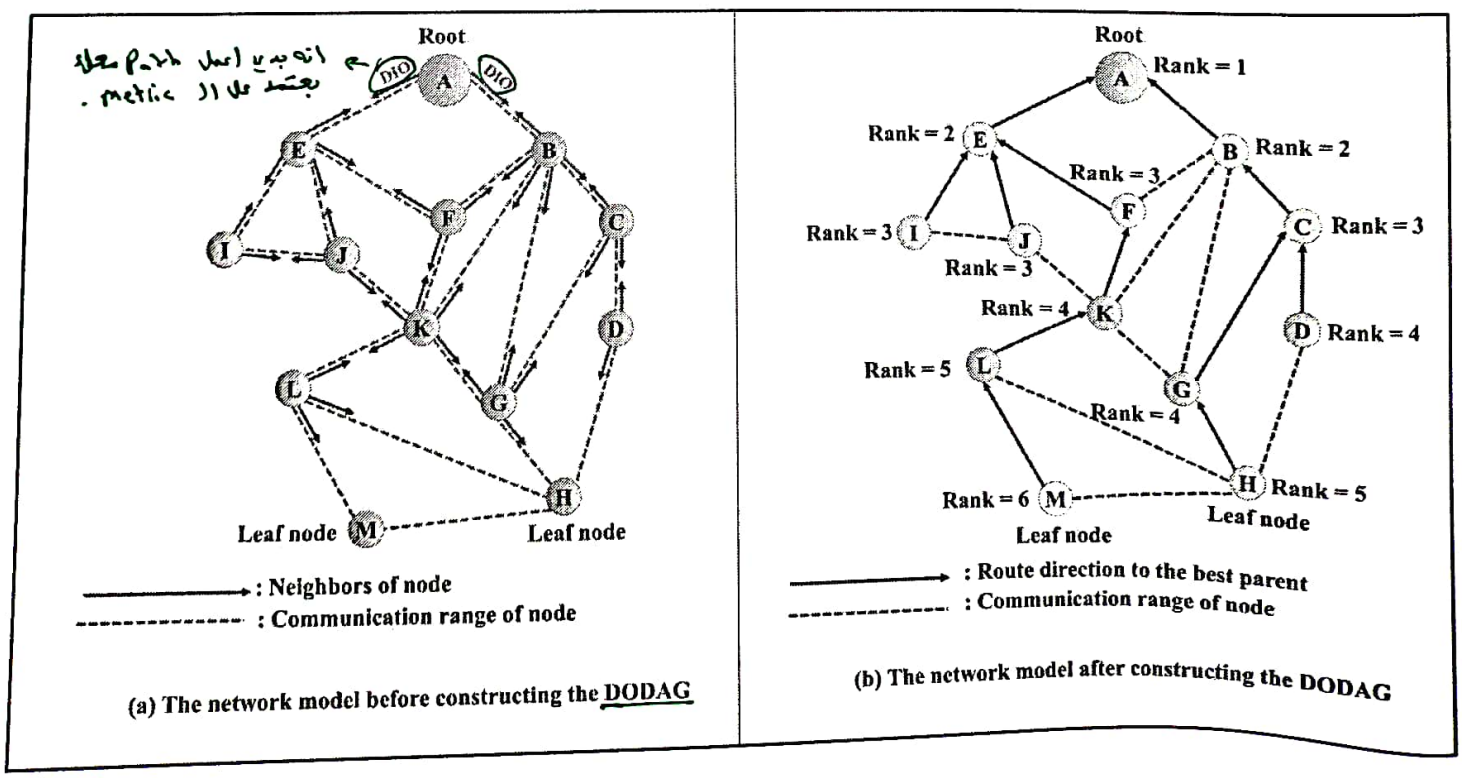


Figure B.1: The network topology and nodes distribution of before and after constructing the DODAG