Chapter 3

WSNs & Energy Efficient Routing

In this chapter, Section 3.1 gives a background on the challenges raised by WSN, together with the strategies that can be used at the network level to improve the energy-efficiency of environmental monitoring applications. The remainder of this chapter is structured as follows. Section 3.1.1 provides an overview of the current technology, and illustrates the constraints of wireless sensors in terms of computational, network throughput and energy resources. Section 3.1.2 describes the main actors of environmental monitoring with WSN, and introduces the issues related to networking and interfacing WSN. Section 3.1.3 emphasizes one of the main characteristics of WSN, called data-centric networking. Section 3.1.4 finally presents optimization strategies, which can be used to reduce energy consumption in environmental monitoring tasks, and in particular describes aggregation services. Section 3.2 onwards discusses various methods to improve energy efficiency before discussing a sensor specific MAC protocol. Finally a discussion on Low Energy Aware Clustering Hierarchy (LEACH) is completed and simulation results of each the protocols are also presented.

3.1 Preliminaries Wireless sensor networks

The purpose of a W-sensor network is to monitor a physical phenomenon by gathering and delivering information to the intended party. Sensor nodes are deployed into a particular area and perform certain tasks of sensing and tracking, and convey information to a base station. The successful operation of such tasks is dependent on the nodes ar}

& dielvng infrmnn to the intrstd prnty. Snsr nouds ar deploued intoae aa pntcr fd & prfrm certeain tsks aaf snsng aur trckng & conveiye infrmn toa aa b/s. Th succouss aaf thaisee opratios caan bee
attributed to recent developments in micro-extronic and mechanical systems (MEMS), processors, radars, and memory technologies. Eventually, the information being perceived by the motes within the WSN that's the supply should be transmitted to a sink (control station or BS) from wherever a question was generated and might access the information. There square measure variety of potential models for these small sensing element networks. During this analysis, the components about small sensing element networks are:

- Homogeneity and energy forced of all the nodes also are assumed. In our analysis, so one purpose is to perform simulation of wireless sensing element networks victimization LEACH and verify the energy savings. Henceforth this could be compared with energy savings achieved through cc techniques.

Whilst the evolution of wireless sensing element networks was motivated by military applications like track surveillance, Nowadays such networks are used in several industrial and shopper applications like process observation & management, machine health observation etc.

### 3.1.1 Sensor technology

A wireless sensing element, or sensing element node, could be a device usually composed of a microchip, a memory, a radio transceiver, an influence supply and one or additional sensors [220]. In several WSN applications, it's necessary that the measurements are geolocalized.

When sensors are randomly deployed, sensor nodes may also feature a geopositioning system (GPS) to obtain the location information. The schematic of basic wireless sensor network devices is represented in Figure 3.1. The current generation of commercially available wireless sensor hardware has the size of a
small wallet, and is designed mainly for experimental research purposes. The next generation of wireless sensors is well illustrated by the seminal smart dust project [223], which took place at the University of Berkeley between 1998 and 2001, and which led to the design of a laboratory prototype wireless sensor whose volume was about 5mm³. The design of viable millimeter scale wireless sensors continues to be the subject of research efforts. The resources available on a sensor node inevitably depend on its size, and the smaller the size more constrained it becomes.

**Figure 3.1**: Schematic of a basic wireless sensor network device [221].

<table>
<thead>
<tr>
<th></th>
<th>WINS NG 2.0</th>
<th>MicaDot</th>
<th>Smart Dust</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost</td>
<td>$100s</td>
<td>$10s</td>
<td>&lt;$1</td>
</tr>
<tr>
<td>Size (cm³)</td>
<td>5300</td>
<td>40</td>
<td>.005</td>
</tr>
<tr>
<td>Weight</td>
<td>5400</td>
<td>70</td>
<td>.002</td>
</tr>
<tr>
<td>Battery capacity</td>
<td>300</td>
<td>15</td>
<td>(Less)</td>
</tr>
<tr>
<td>Sensors</td>
<td>Off-board</td>
<td>Integrated on PCB</td>
<td>MEMS</td>
</tr>
<tr>
<td>Memory</td>
<td>32MB RAM</td>
<td>4KB RAM</td>
<td>(Less)</td>
</tr>
<tr>
<td>CPU</td>
<td>400 MIPS</td>
<td>4 MIPS</td>
<td>(Less)</td>
</tr>
<tr>
<td>Radio range</td>
<td>100m</td>
<td>30m</td>
<td>(Less)</td>
</tr>
<tr>
<td>Operating system</td>
<td>Linux</td>
<td>TinyOS</td>
<td>(Smaller)</td>
</tr>
</tbody>
</table>

**Table 3.1**: Comparison of the resources available on sensor nodes as the size
The need for experimenting WSN protocols, algorithms and applications in real world conditions has led to the design of wireless sensor platforms for research. The University of Berkeley initiated such designs in 1999 with the WesC, a platform that approximated the functionalities envisioned by the Smart Dust project. This design was followed by the MICA, MICAz, MICA2DOT, and TelosB platforms, which have been the most widely used sensor nodes in academia for prototypical WSN deployments. The characteristics of these sensor nodes, also known as motes, are summarized in Figure 3.2. These nodes use components-off-the-shelf (COTS) hardware instead of integrated silicon designs in order to allow easy customization of the boards, and to reduce the production costs. Following the Smart Dust vision, these platforms run 8-bit microcontrollers and have a few tens of kilobytes of memory. Low data rate radios, ranging from a few tens to a few hundreds of kilobits per second, enable the communication. Sensors are either integrated (such as on the dot 2000 or the TelosB), or attached by means of a daughter board. Their size essentially depends on the batteries, typically a pair of AA cells.

The integration of these COTS platforms in silicon would reduce their size to a few millimeter cubes, as illustrated by the Spec platform, the silicon integrated counterpart of the MICA platform in the Smart Dust project. Such small-scale wireless sensors are however still not viable, as a number of issues in hardware robustness and communication protocols must be further investigated.

Among current platforms, the MICA motes and the Telos are nowadays the most popular ones for WSN prototyping. The Telos outperforms the MICA motes, particularly in terms of radio throughput and power consumption. The Telos
architecture features lower power electronics for the flash memory and the microprocessor, and the radio rate increased from about 40 kbps to 250 kbps. Despite these advantages, the earlier appearance of the MICA motes family on the market, and the relatively small differences in functionalities between the MICA motes and the TelosB led a large number of laboratories to adopt MICA motes for experiments.

![Figure 3.2](image)

**Figure 3.2:** The family of Berkeley motes and their capabilities [222].

The normal hardware elements of a device node embody a radio receiver, an embedded processor, internal and external memories, power supply & power autonomy.

### A. Embedded Processor

In a detector node, the practicality is associated with the embedded processor to schedule tasks, manage information & management.

The kinds of embedded processors which will be utilized in a detector node embody Microcontroller, Digital Signal Processor (DSP), Field Programmable Gate Array (FPGA) & Application Specific Integrated Circuit (ASIC). Among these alternatives, the microcontroller has been the foremost used embedded processor for detector nodes thanks to its flexibility to attach to alternative devices & its low-cost value. For instance, the most recent CC2531 development provided by Chipcon (acquired by Texas Instruments youses 8051 microchip & also the Mica2 molecule platform provided by bow youses ATMega128L microcontroller.

B. Transceiver

A transceiver is chargeable for wireless communication as a sensing element node. The assorted decisions of wireless transmission media embrace Reedio Frequency (RF), optical device & Infrared. ARF primarily based communications phits to the most of WISEN applications. The prepared states of a receiver area unit send, recv, eye & sleep. Mica2 molecule youses 2 varieties of RF radios: ARFMR1000 & Chipcon CC1000. The outdoor transmission vary of typical Motes is around one hundred fifty meters.

A. Memory

Memories in a very sensing element node embody internal nonvolatile storage & RAM as a microcontroller & external nonvolatile storage. As an example, the ATMega128L microcontroller running on Mica2 molecule has 128-kiloB flash program memory & 4-KiloB static RAM. Further, a 4-MegaB Atmel AT45DB041B serial flash chip will offer external recollections for translucent substance & Mica2Motes (McGraw, 2003).
B. Power supply

In a sensing element node, power is needed for sensing, communication and processing. Most energy is needed for electronic communication than for sensing and processing. Power may be kept in batteries or capacitors. Batteries are the first supply of power offer for sensing element nodes. As an example, Mica2 corpuscle runs on a pair of AA batteries thanks to the restricted capability of batteries minimizing the energy consumption is usually a key concern throughout WSN operations. To get rid of the energy constraint, some initial analysis, performing on energy-harvesting techniques for WSNs has additionally been conducted. Energy-harvesting techniques convert close energy (e.g. solar, wind) to current and therefore the aim is to revolutionize the facility offer on sensing element nodes. A lot of on energy-harvesting sensing element nodes may be found in very textbooks.

C. Sensors

A sensing element could be a hardware device that produces a measurable response signal to an amendment in a very shape like temperature, pressure and humidity. The continual Analog signal detected by the sensors is digitized by an analog-digital converter and sent to the embedded processor for any process. As a sensing element node could be a micro-electronic device hopped-up by a restricted power supply, thus hooked up sensors ought to even be little in size and consume extraordinarily low energy. A sensing element node will have one or many varieties of sensors integrated in or connected to the node.

D. Operational System

The role of any OS (OS) is to push the event of reliable application package by
providing a convenient and safe abstraction of hardware resources. OSs for WSN nodes is often less complicated than general OSs each as a result of the special necessities of WSN applications and since of the resource constraints in WSN hardware platforms. TinyOS is maybe the primary OS specifically designed for WSNs. It options a component-based design that allows fast innovation and implementation whereas minimizing code size by the severe memory constraints inherent in WSNs. small OS’s part library includes needed network protocols, distributed services, drivers, and knowledge acquisition tools - all of which may be any refined for a custom application totally different from most alternative OSs, TinyOS is predicated on AN event-driven programming model instead of multithreading.

TinyOS programs are composed into event handlers and tasks with run-to-completion linguistics. Once an external event happens like an incoming knowledge packet or a sensing element measure, TinyOS calls the acceptable event handler to handle the event. Event handlers will record tasks that are scheduled by the TinyOS kernel at later stage.

3.1.2 Environmental Monitoring

The intent of this thesis is to employ a data collection for environmental monitoring, where the task consists in retrieving the measurements from the sensor network at regular time intervals. This would be the scenario for subsequent analysis and research of the thesis. In a typical environmental monitoring scenario, four main entities interact [226]:

The phenomenon: It is the entity of interest to monitor. The phenomenon may be the vibration patterns on a bridge, the location and size of a fire in a forest, or the number and type of animals present in an environment. The data collected about the
phenomenon may be analyzed/filtered by the sensor network before being communicated to the base station/ the observer.

**The observer:** It is the end user or the application interested in obtaining information collected by the sensor network. The observer may formulate queries to the network and receive responses to these queries by means of the b/s.

**Thye snsr ndes or sensors:** They are the wireless device that implement the physical sensing of an environment and the reporting of the measurements to a base station. The sensors may be placed anywhere in the sensor field, according to predetermined positions, or randomly deployed, from an airplane for example. Once deployed, sensors are assumed to be static, and to run unattended with an on renewable amount of energy.

**The base station:** It is assumed to possess higher resources than sensor nodes, and provides centralized data storage and processing unit for the data collected by the sensor network. It may connect the WSN to the Internet.

The design of data collection systems for sensor networks has been addressed early in the research literature [221]. In particular, the three following aspects are of significant practical importance. First, an observer must be able to communicate with the network in order to retrieve data. In data collection tasks, it is in particular desirable that the user can select what measurements to retrieve, change the sampling frequency, or address a query to a specific region of the network. Second, once the observer has formulated a query, it must be communicated to the network, and sent to the appropriate nodes. Finally, once all the sensing element nodes concerned within the question are reached, the question should be dead in such a way that the observer retrieves the requested data.

The collection of data from the sensors to the b/s requires in most cases the
establishment of a routing structure. The radio range of sensors is limited, and therefore intermediate sensors must relay the data from sensors distant to the control. The transmission of data using multiple relay nodes is called multi-hop routing. In data collection tasks, a typical routing structure is a tree connecting all the nodes to the base station [225, 226].

An example of such a tree, also known as data gathering tree, is given in Figure 3.3. The dotted circle illustrates the radio range of the dark gray sensor. The light gray sensors lying inside the circle form its neighborhood, i.e., the set of other sensors it can communicate with.

![Figure 3.3 Data gathering tree enabling communication between the device nodes to the bottom station.](image)

### 3.1.2 Data-centric paradigm

At the conceptual level, one of the fundamental differences between wireless sensor networks and other networks is that the routing and querying techniques can be made more efficient if the communication is based directly on application specific data content instead of the traditional IP-style addressing [221]. This focus on data
content is referred to as the data centric paradigm in [227], whose main characteristic is the routing based on attributes. In attribute-based routing, the sensor nodes may be identified not on the basis of a network address, but on the basis of their attributes, i.e., the pieces of information they hold. These attributes include for example the location or the measurements of a sensor node. A reference approach that uses this feature is the Directed Diffusion technique [228].

In Directed Diffusion, both the observer’s queries and sensor’s attributes are described through sets of attribute-value pairs. Thus, an observer may request sensor nodes within a given region to report their temperature measurements every thirty seconds for one hour by defining the set following attribute-value pairs:

```plaintext
Type=temperature  //Type of measurements required
Location=[0.0, 15, 35]  //Coordinates of the region of interest
Duration=10:00:00 //Duration of the query
Frequency=30 //Sampling frequency
```

The query is sent throughout the network, and executed by all sensor nodes that are located within the region and that have a temperature sensor. An example of response to the query can be:

```plaintext
Id=324 // Node unique identifier
Type=temperature //named record type
Value=25.3°C //value of this type
Location=[10,25] //location of the measurement
Time=04:23:30 //time of measurements
```

The nodes may have identifiers, but these do not require being part of the observer’s request. Directed diffusion is organized in three stages (Figure 3.4):
Figure 3.4: Main stages of Directed Diffusion [228]. The sensor node that matches the query is node 1.

1. **Query dissemination**: When the observer is interested in collecting data, it sends through the base station a query that specifies the type of data required. Each node, on receiving the query, rebroadcasts it to its neighbors. Figure 3.4(a) illustrates this stage, with node 7 transmitting to nodes 6 and 5, which themselves reach nodes 4, 2, and 3, up to node 1, which matches the query.

2. In addition, as the query is disseminated, each node sets up an interest gradient to the nodes from which the query was received. The gradient quantifies how efficient the different nodes are at routing the query. The way the gradient is computed is application specific, and aims at minimizing a certain cost, such as the number of hops or the traffic load on a given path.

3. Data are then transmitted from node 1 to the base station using the path that has the best gradients. Multipath delivery can be used in order to make the message delivery more robust to transmission failure.

The use of attributes to match sensor nodes and observer’s interests make sensor networks conceptually close to databases. In classical database management systems,
data is accessed by queries from users or applications, which specify information to retrieve in a high level language such as SQL. The SQL syntax has therefore been introduced in the WSN domain very early, and has been extended to meet some of the specificities of WSN, in particular by adding the clauses related to the frequency and duration of the data collection [226, 229, 230]. Using SQL, the query formulated above in terms of attribute-value pairs becomes

```
SELECT temperature FROM sensors
WHERE location=[0,0,15,35]
DURATION=00:00:00,10:00:00
EPOCH DURATION 30s
```

The fact that the SQL syntax is widely known thus makes the interfacing between observers and sensor networks more natural and simple.

### 3.1.4 Energy efficiency and aggregation services

Among the sensor node resources, energy is widely considered as the most precious resource as it determines the lifetime of an application. In full active mode, the lifetime of a sensor node such as a T-mote is only a few days. Improvements in battery design and energy harvesting techniques only offer partial solutions to extend sensor nodes’ lifetime, and therefore most of the WSN literature has focused on the design of protocols with energy efficiency as the primary goal [221].

Energy consumption depends on the amount and type of activities performed by sensor nodes. In order to more precisely control energy consumption, sensor nodes are usually designed so that their different components can be powered on and off. The current consumption is the lowest in the standby mode, in which all components are switched off except the clock, and is on the order of a few microamperes. The use of the microcontroller unit (MCU) typically requires a few milliamps, and is therefore three orders of magnitude higher than the standby mode. The additional use of the radio or the memory component increases by an order of magnitude the overall current draw.
This is illustrated in Table 3.2, which details the current draws of the MICA2, MICAZ and Telos motes for different modes of operations.

**Table 3.2:** Comparison of the current consumption of MICA2, MICAZ and Telos mote [231].

<table>
<thead>
<tr>
<th>Operation mode</th>
<th>MICA2</th>
<th>MicaZ</th>
<th>Telos</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standby</td>
<td>19.0 µA</td>
<td>27.0 µA</td>
<td>5.1 µA</td>
</tr>
<tr>
<td>MCU Idle</td>
<td>3.2 mA</td>
<td>3.2 mA</td>
<td>54.5 µA</td>
</tr>
<tr>
<td>MCU Active</td>
<td>8.0 mA</td>
<td>8.0 mA</td>
<td>1.8 mA</td>
</tr>
<tr>
<td>MCU + Radio RX</td>
<td>15.1 mA</td>
<td>23.3 mA</td>
<td>21.8 mA</td>
</tr>
<tr>
<td>MCU + Radio TX (0dBm)</td>
<td>25.4 mA</td>
<td>21.0 mA</td>
<td>19.5 mA</td>
</tr>
<tr>
<td>MCU + Flash Read</td>
<td>9.6 mA</td>
<td>9.4 mA</td>
<td>4.1 mA</td>
</tr>
<tr>
<td>MCU + Flash write</td>
<td>21.6 mA</td>
<td>21.6 mA</td>
<td>15.1 mA</td>
</tr>
</tbody>
</table>

A simple solution to save energy & extend the lifetime of a sensor node is to operate the node with periodic switching between standby and active modes. This approach, called duty cycling, is particularly suitable for periodic data collection tasks [220, 221, 232], where the operation of the sensors is by definition periodic. Since the collection and the transmission of one packet is typically less than one second, the energy savings can be significant if the frequency of the data collection is low. The sampling of one measurement every minute can for example extend the lifetime of a node by a factor of at least sixty using the duty cycling.

In multi-hop networks, the nodes must be synchronized so that parents and children in the routing tree are active at the same moment. Different systems, such as TAG [226, 233], Cougar [234] or Dozer [225] for example, have been proposed to carry out the synchronization in a time & energy-efficient way. TAG is the most well known representative of such systems.

TeAyeG stands for Tny Aggregtn & iss ann in-networking and aggregtn srvc fr snsrtwrks tht haz bean implmntd inn Tny OpSy [235], ana oprtng systm wth
a low memory footprint specifically designed for wireless sensors. TAG allows to aggregate data within a WSN in a time and energy conserving manner. To that end, an epoch is divided into time slots, in such a way that the activities of the sensors are synchronized as a function of their depth in the routing tree. Any algorithm can be used to design the routing tree, as long as (i) it allows the data to flow in bidirectionally in the tree, and (ii) it avoids sending duplicates [226].

The purpose of TAG is to minimize the amount of time spent by sensors in powering their different components and to maximize the time spent in the standby mode, in that all electronic components are switched off. This synchronization allows to significantly extending the lifetime of the sensors. An illustration of the activities of the sensors during an epoch is given in Fig. 3.5, for a network of four nodes with a routing tree of depth three.

(a) Routing tree of depth three. (b) Activities carried out by sensors depending on their level in the routing tree (adapted from [226])

Figure 3.5: Multi-hop routing along a routing tree, and node synchronization for an efficient use of energy resources
The establishment of the routing tree works in a similar manner as Directed Diffusion described in Section 3.1.3. The observer’s query is flooded from the base station to the sensors, which identify at the same time a possible routing tree. Each node chooses another node as its parent, using a metric such as the minimum number of connections. Nodes also synchronize their clocks by adding timestamps in the messages, and define their own activity schedule such that their radio transmission period is within the radio reception period of their parents. In order to overcome limitations in the quality of the clock synchronization between parents and children, the parents listen for longer than the transmission interval of their children. The duration of the interval in which parents receive the values from their children needs to be long enough so that all children can report, but not so long so that nodes deep in the tree can report their value before the end of the epoch.

Aggregated services such as TiAG allow to both reducing energy consumption by carefully scheduling sensor node’s activity, & by allowing the measurements to be aggregated as they are received by the base station. This aggregation allows reducing the number of packet transmissions, and can further be used to compute parameters in a distributed manner.

3.1.3 Summary

Wireless sensor networks are strongly resource constrained, particularly in terms of energy. In environmental monitoring applications, where the network is expected to run unattended for months or even years, the need for energy-efficient data collection schemes has driven the design of new routing and querying strategies. In particular, the optimization problems resulting from maximizing the sleeping time of sensor nodes
have led to the design of aggregation services.

The strategies presented in this section however did not take into account the fact that sensor network data are often correlated over space and time. Given that the microprocessor consumes one order of magnitude less energy than the radio module, in-network processing strategies can be designed to detect these correlations, and to allow further energy savings by removing the correlations within the network. Learning techniques are in this respect among the most promising approaches.

3.2. Energy-Efficient Routing

It is currently possible to fabricate ultra-small detector nodes that may be scattered over a district of interest to collect info [237]. The events detected by these nodes have to be compelled to communicate to gateways or users World Health Organization faucet into the network. This communication happens via multi-hop routes through different detector nodes. Since the nodes have to be compelled to be unobtrusive, they need a little form-factor and thus will carry solely a little battery. As a result, they need a restricted energy provide and low-power operation should be done. Multi-hop rtng prtcls fr thiez ntwrks essentially neid to bee designed with a spotlight on energy potency.

Various enrgy efficnt rtg protcls hve bean surveyed inn Chapter 2. Here, we present two specific protocols - section 3.2.2. covers a specific media access issue for sensor networks- S-MAC Protocol and in the next section we describe a Low Energy Adaptive Clustering Hierarchy (LEACH) protocol.

3.2.1. Improving Energy-Efficiency

Ad-hoc routing algorithms concentrate on avoiding congestion or maintaining property once sweet-faced with quality [238] for energy economical routing. They are doing not
think about the restricted energy provision by the network devices. The instance of figure one illustrates however the restricted provide alters the routing issue. Nodes A and E initial send fifty packets to B. Afterwards, F sends one hundred packets to B. From a load leveling perspective the well-liked methods measure ADB, ECB and FDB are several amongst many.

However, once the nodes measure energy affected specified they'll solely send one hundred packets, these methods are not any longer optimum. Indeed, D would have ran down five hundredth of its energy before it will forward packets from F to B. during this case, all packets might are delivered by selecting methods ACB, ECB and FDB. If, rather than F, node C would became active, A ought to have used the first path ADB.

This simple case study highlights the subsequent crucial observation: optimum traffic programing in energy affected networks needs future information. In the example, a most variety of packets will reach B on condition that right from the beginning it is

Figure 3.6: Load versus energy orienting routing
known specifically once (and which) nodes can generate traffic within the future.

Ideally, it might just like the sensing element network to perform its practicality as long as getable. Optimum routing in energy affected networks isn’t much possible (because it needs future knowledge). However, we are able to soften our necessities towards a statistically optimum theme that maximizes the network practicality thought of over all potential future activity. A theme is energy economical (in distinction to ‘energy optimum’) once it’s statistically optimal and causative (i.e. takes solely past and gift into account).

In most sensible surroundings watching applications, we have a tendency to don’t need any coverage gaps to develop. We have a tendency to thus outline the period of time we wish to maximize because the worst-case time till a node breaks down, rather than the common time over all things. However, taking into account all potential future situations is just too computationally intensive, even for simulations. It’s thus actually impracticable as a tenet to base sensible schemes on.

It is thus actually impracticable as a tenet to base sensible schemes on. To derive a sensible guideline, we have a tendency to begin from the subsequent observation: the minimum hop methods to a user for various streams tend to own an outsized variety of hops in common [239]. Nodes on those methods die sooner and so limit the period of time of the network. Figure 3.7 presents a typical energy consumption bar graph at a precise point in time. Some nodes have hardly been used, whereas others have virtually fully drained their energy.

As nodes that square measure running low on energy square measure a lot of vulnerable to die sooner, they need become a lot of crucial. If we have a tendency to assume that
each one the nodes square measure equally necessary, no node ought to be a lot of crucial than the other one. At every moment each node ought to thus have used regarding a similar quantity of energy that ought to even be decreased. The bar graph of figure three.8 is so a lot of fascinating than the one amongst figure three.7, though the whole energy consumption is that the same. Attempt for a compact energy bar graph interprets into the rule that traffic ought to be contact the network as uniformly as potential. Since visualizing the bar graph over time is tough, we have a tendency to might use the foundation mean sq. ERMS as associate indicator instead (the lower this price the better). It provides data on each the whole energy consumption and unfolds.

Figure 3.7: Undesirable energy histogram
3.3. S-MAC Protocol

Like altogether shrd-med NWs, med acc management (MAC) is a very imp tech that allows the triple-crown op of the WSN. One elementary tsk of the Macintosh prtcl is to avd coll so 2 officious nodes don't xmitt at the same time. Thr square measure large Macintosh prctls tht are devp for wrls vce and electronic commn NWs. Typ eg embody the Tiam Divzn Mltpl Acsez (TDMA), Czode Divzn Mltpl Access (CDMA) & cntn-bsd prtcls such as IEEE 802.yy family [237].

To design a decent Macintosh protocol for the wireless sensing element networks, it is to think about the subsequent attributes. The foremost is that the energy effectively. As declared higher than, sensing element nodes square measure probably to be battery
hopped-up, and it’s generally terribly troublesome to alter or recharge batteries for these motes. In fact, in future we have a tendency to expect some motes to be low-cost enough that they are discarded instead of recharged. Extending NW period of time for these motes could be a crucial issue. Another necessary attribute is that the quality to the modification in NW size, mote density and topology. Many motes could die over time; some new motes can be a section of it; some motes could move to all completely different locations.

The configuration changes over time what is more thanks to many reasons. A decent Macintosh protocol ought to simply accommodate the said NW changes. Alternatively important attributes embrace fairness, latency, through place & spectrum utilisation. These attributes square measure typically the most considerations in ancient worlds vce and knowledge NWs, however in sensing element NWs they are secy.

The section presents sensor-MAC (S-MAC), a typical Macintosh protocol expressly designed for worlds sensing element NWs. While reducing consumption is the main goal within the style the main goal within the style, the protocol jointly has sensible measurability and collision dodging capability. The protocol achieves sensible measurability & collision dodging by utilising a comb programing & rivalry theme. To attain the required goal of energy potency, like to acknowledge what square measure the most sources that consume inefficient use of en; additionally as well as trade-offs are to build to cut back energy consumption.

*Rsns of En Wste*

When a receiving mote receives more than one packet at the same time these packets are called "collided packets" even when they collided partly. Collisions increase energy consumption.
This occurs as all packets that have collided need to be discarded and re-transmitted. It is possible to recover some packets by capzur effect, but a nmbr of conditions have to be met for its success.

All nodes receive packets, even though they are not meant for them. This occurs due listening a second energy wasting reason. Also a third reason is the management of packets causing energy waste. Least variety of mgt pkts ought to be accustomed and create a knowledge xmission. Concretely th the main srces of en wste is inactivity listn, or being attentive to associate degree idl ch to rx attainable tfe.

Over-transmission is the laist reazon for enrgy waste. O/T iz the xmission aaf the message when the terminus hub is not prepared. In view of the foregoing a good MiAC ought to keep these vitality wastage.

Commn Ptnrs

In [240] three communication models are described of WSNs: i) broadcast ii) Converge Cast iii) local gossip. Type (i) is generally used by a base station to transmit same data to all SNs of the NW. Queries of sensors; query processing architecture or program updates are typical broadcasts of a base station and sent regularly to the entire network. Packets are also broadcast so it should be clearly differentiated with communication broadcast.

Fr the published kind communication, all motes of the NW are meant rxs whrs fr the packet th meant rxs are the ndes inside th commn vary of the transmittal transmittal node. In sme situations, the snrs th notice associate interloper
communicate with one another domestically.

This sort of correspondence example is called nearby gossip, where a sensor sends a message to its neighboring hubs inside a reach. The sensors that recognize the intruder, then, need to send what they see to the data focus. That correspondence example is called unicast, where a gathering of sensors convey to a particular sensor. The objective hub could be a group pioneer, information combination, focus b/s.

In conventions that incorporate grouping, group pioneers speak with their part what’s more accordingly the planned recipients may not be all neighbors of the group head, yet simply a subset of the neighbors. To serve for such situations, characterize a fourth sort of correspondence design, multicast, where a sensor makes an impression on a particular subset of sensors.

3.3. Components of S-MAC Protocol

These are based on the design criterion of reducing energy wastage. Recall these are idle listening; collision; overhearing and packet management overhead. It is essential to summarize our assumptions regarding the wireless sensing element network and its applications, before explaining the components of S-MAC. To conserve energy multi-hop communication model is adopted as sensor networks incorporate massive quantities of motes. Most comm ns can occur between nodes as pairs instead of to one BS. In-network process is essential to NW period [244], & implies that knowledge are processed as whole messages in near

near locations and so forwarded. Pkt or frag-level intrvl fm mltp srcs solely will solely will increase ovrl lat. Fnlly, tend to expect that applications can have long idl pds & may
may tlrte lat on th ordr of NW electronic communication tme.

A. Prdc Lstn & Slp

Due to the character of WSNs, nodes square measure idle most of the time therefore nodes sleep and wake sporadically. In SMAC to conserve energy because the rte is extremely lo throughout tns era, it's not essential to stay ndes lstn all th tme.

Periodic sleep wake cycle reduces Idle Listening.

B1The fundamental theme

The basic theme shown in Figure 3.9 is that every node sleeps sporadically, and so awakens up, listens to imagine if the other hub yearnings to identify with that. All through resting, the he hub turns off its radio, and sets a clock to conscious it later. A whole cycle of listen and slumber is named frime. The amount of your time for listening and sleeping are often elite in step with completely different eventualities. For simplicity these values sq measure a similar fr all the ndes. Our theme needs per synch amng ngbring ndes to rmdy thr clk drft. Intend to use 2 ways to form it face up to in synchronizing errs. Frst amongst nodes all timestmps that square measure changed square measure relative instead of abs. Sec, th lstn span is considerably Ingr thn clk err or drft.

Figure 3.9: Periodic listen and sleep.
Thus listen length of zero.5s is over one hundred and five times long than typical drift rate. Compared with TTDMA schemes with terribly short time slots, our strategy needs additional relaxed synchronization among neighboring nodes. All nodes can decide on their listen/sleep schedules. However, to attenuate management overhead, we have an inclination to love neighbors to synchronize. That is, they listen at identical time and attend sleep at identical time. It needs to be detected that not all neighbors can synchronize on multi-hop networks. Two neighbors A & B might need full schedules if they ought to synchronize with different nodes, C and D, severally.

Exchange of Schedules - Nodes do this by a broadcast mechanism to any or all immediate neighbors. The broadcast allows all neighbors to ensure they have a different schedule. In case more than one neighbor is transmitting to the CH, access control is required. This control for media is similar to IEEE 802.11 requiring exchange of aRTeS (Rqst Toa Snd) and CeTS (Clr Toa Snd) packets.

The hub that initially conveys the RTS bundle wins the medium, and conjointly the beneficiary will answer with a CTS parcel. When they start information transmission they concede their slumber plans until they complete transmission. An alternate normal for the develop is that it structures hubs into a level topology. Neighboring hubs will decide to visit each other notwithstanding what listen plans they require. Synchronous hubs are a virtual bunch however there's no commendable group thus no issues of between bunch interchanges and obstruction. This subject is kind of clear to adjust to topology changes.
The drawback of the theme is that the latency is accrued because of the periodic sleep of every node. Moreover, the delays get additive on every hop, therefore the latency demand of the applying places an elementary limit on the sleep time.

B2 selecting & Maint Sch

A node has to select a schedule of exchange of data before starting its periodic sleep and listen with its neighbors. All nodes maintain a schedule that stores the schedules of all its noted neighbors. It follows these steps below to pick out its schedule and maintain its schedule table.

1. Choosing Sleep Time & Synchronization -

   This task is done by a node called a synchronizer since it selects its schedule. It does by an arbitrary transmission of a SYNCH message that would go to a sleep after 't' seconds.

   node a synchronoscope, since it selects its schedule.

2. On the other hand if a node receives a schedule from some other node before its own schedule, it follows that schedule. In these circumstances we have a follower node. But this node re-broadcasts its 'SYNCH' message indicating that it will sleep in 't - td' seconds. "The random delay is for collision dodging, in order that multiple followers triggered from a similar synchronoscope don't systematically collide once rebroadcasting the schedule."

3. If a node receives a special schedule when it selects its own schedule, it adopts each schedule (that is, it schedules itself to get up at the days of
each is nghb & itself). It B/Cs it own sch bfr getting to slp.

That nodes adopt multiple schedules is an exception. Generally every node follow existing schedules bfr flwing an indep freelance one. On th othr hnd, psbl ths nghb ndes fl to discvr ea other at starting because of due to collns when b/c thr sch. Thy may still fnd ea other ltr in th subsqnt prd lstn.

To ills th algo, cnsdr a NW whr all nde c an hr ea other. Th tmr of one nde will fr frst & its b/c will synch all of its prs on its sch. If instd two ndes indel assgn sch (either bec thy cnnt hr ea other or bec thy hppn to xmit at nrly th same tme), ths ndes on brdr betwn the two schs will adpt both. In thy wy, a node cn ony nds to snd once for a b/c pkt. The disadvtg is tht thse brdr ndes hve lss tme to slp & cnsme mre en thn othrs. "The disadvantage is that these border nodes have less time to sleep and consume additional energy than others."

Anthr opt is to lt the ndes on the brdr adpt solely the primary received schedule. Since it is aware of the opposite schedules that other neighbors have adopted, it will still visit thm. Hwvr, for b/c pkts, it has to repeat th 2 completely diff schs. Th advtg is tht the brdr ndes hv a similar straightforward ptrtn of amount of lstn & slp as different ndes.

The listen/sleep theme needs "synchronization among neighboring nodes though the prolonged listen time will tolerate fairly massive clock drift, neighboring nodes still got to sporadically update every other schedule to forestall long-time clock drift. The changed amount is often quite long. The measurements on typical workplace nodes show
that they are often on the order of tens of seconds."

C. Collision and Overhear shunning

Collision shunning may be a primary goal of media access protocols. S-MAC adopts a cntn bsd prctl. It is traditional that everyone its neighbors receive any packet transmitted by a node despite the fact that just one node is needed to receive the packet. "Overhearing makes contention-based protocols less economical in energy than TDMA protocols. So it has to be avoided."

C.1 Colln shunning

As can be seen from the previous description of maint lstn & slp cycles ocassions occur when several senders want to transmit to a rx. Whilst 802.11 does a good job of colln avoidance, the prctl under study has smpl prcdrs incl bth virtl & phy carr snse & RTS/CTS exch. It also has a good mech to handle th hddn trmnl drawback [258].

In the length field of every transmitted packet there's a sign of the remaining coordinated universal time. Therefore if a node receives a packet meant for one more node, it is aware of the length it's to stay silnt. Th nde recs ths worth in associate var named th Ntwk Allcn Vctr (NAV) [259] & sets a tmr for it. On evry occasion once th NAvectr tmr frs th nde dec the NAvectr till it rchs zro. Once a nde has knowledge knowledge to send, it initial examines th NAvectr. If its worth isn't zro, th nde dtrmns tht th med is bsy. Ths method is named virtl carr snse. Phy carr snse is prfmd at th phy lyr by being attentive to the ch fr attainable transmission activity.

Th mthd was delineated earlier. The irregular carr snse tme is extremely necessary necessary fr colln shunning. Th med is decided as free if each virtl & phy carr snse indicates tht its fre.
"All senders perform carrier sense before initiating a transmission. If a node fails to access the medium, it goes to sleep and wakes up once the receiver is free and listening once more. Broadcast packets square measure transmitted while not victimisation RTS/CTS. Unicast packets follow the sequence of RTS/CTS/DATA/ACK between the sender and therefore the receiver."

C.2 Ovrhrg shunning

"In 802.11 every node keeps being attentive to all transmissions from its neighbors so as to perform effective virtual carrier sensing. As a result, every node overhears plenty of packets that don't seem to be meant for it." This is often a major waste of energy, additional therefore once kind issue that's node density is high and traffic load is serious.

"The S-MAC protocol tries to avoid overhearing by rental meddlesome nodes attend sleep when they hear associate RTS or CTS packet. As knowledge packets square measure ordinarily longer than management packets, the approach stops neighboring nodes from overhearing long knowledge packets and therefore the following ACKs. In next section we tend to describe a way to with efficiency transmit a protracted packet combining with the overhearing shunning. Currently we glance at that nodes ought to attend sleep once there's an energetic transmission happening."

As shwn in Fig three.10, nde A, B, C, D, E, and F foms a mltp-hp NW nde wherever every will solely hr th xmissions from its immdt ngbhs. Spse nde A is presently sending a knowledge pkt to B. the difficulty is, that of th rem ndes ought to currently attend slp.
Figure 3.10: Who ought to sleep once node A is sending to B?

"Note that collision happens at the receiver. It's cleared that node D ought to attend sleep since its transmission interferes with B’s reception." So it is straightforward to point out that node E & F doesn't turn out interference in data exchange between A and B, so that they don't have to attend sleep. Ought to node C attend sleep? C is over one, rather 2-hop "from B, and its transmission doesn't interfere with B’s reception."

Therefore it will transmit to its different neighbors like E. How, C is unable to urge any response from E, except CTS or knowledge, as a result of E’s transmission collides with A’s transmission to node C, therefore C’s transmission is just a waste of energy. In short, all the one-hop neighbors of the sender and therefore the receiver try ought to sleep when they hear the RTS or CTS packet till this transmission is over.

Each node maintains the NAV to point the activity in its neighborhood. Once a node receives a packet meant for different nodes, it updates its NAV by the length field within the packet. A non-zero NAV worth indicates that there's an energetic transmission in its locality. The NAV worth decrements on every occasion once the NAV timer triggers. So a node ought to sleep to avoid overhearing if its NAV is non-zero. It will get up once its NAV becomes zero.

D. Message Passing
Now it is intended to describe a way to with efficiency xmitt a protracted msg in each en & ltny. "Associate data message is that the assortment of reticular units of knowledge. It is often a protracted series of packets or a brief packet, and frequently the receiver has to assemble all the info units before it will perform in-network processing or aggregation."

Th drawbacks of "sending a protracted message as one packet square measure the high price of re-transmitting the long packet if solely some bits are corrupted within the initial transmission. On the opposite hand, if we tend to fragment the long message into several freelance tiny packets, then to pay the penalty of enormous management overhead and bigger delay." It's therefore as a result of the RTS and CTS packets square measure employed in rivalry for every freelance packet.

The S-MAC appch is to frag th lng msg into several manageable frags, & xmitt them in brst. "Just one RTS packet and one CTS packet square measure used. They reserve the medium for sending all the fragments. On every occasion a knowledge phase is transmitted, the sender watches for associate ACK from the receiver. If it doesn't receive the ACK, it'll lengthen the reserved coordinated universal time for an added fragment, and re-send this fragment now."

As bfr, all pkts hv th length fd that is currently th tme required fr sending all th rem knowledge frags and 'ACK 'pkts. "If a neighboring node hears a RTS or CTS packet, it'll attend sleep for the length that's required to transmit all the fragments."

To prevent th hidden terminal drawback 'ACK ' is distributed when every
knowledge fragment. "It's attainable that a neighboring node wakes up or a replacement node joins within the middle of a transmission. If that node is merely the neighbor of the receiver however not the sender, it'll not hear the info fragments being sent by the sender. If the receiver doesn't send ACK overtimes, the new node might erroneously infer from its carrier sense that the medium is evident. If it starts sending, this transmission is corrupted at the receiver."

Every knowledge "fragment and ACK packet conjointly has the length field. During this manner, if a node wakes up or a replacement node joins within the middle, it will properly attend sleep in spite of if it's the neighbor of the sender or the receiver. For instance, suppose a neighboring node receives associate RTS from the sender or a CTS from the receiver, it goes to sleep for the entire message time. If the sender extends the coordinated universal time because of fragment losses or errors, the sleeping neighbor won't remember of the extension now. However, the node can learn it from the extended fragments or ACKs once it wakes up."

It is pertinent to notice "that IEEE 802.11 conjointly has the fragmentation support. We should always imply the distinction between that theme and S-MAC protocol message passing. In 802.11, the RTS and CTS solely earmark the medium for the initial knowledge fragment and therefore the first acknowledgement (ACK). The primary fragment and ACK thenceforth reserves the medium for the second fragment and ACK, then forth. Therefore for each one-hop node, when it receives a fraction or associate ACK, it is aware of that there's an added fragment to be sent. Therefore it's to stay listening till all the fragments square measure sent. Again, for energy-constrained nodes, overhearing by all neighbors exhausts plenty of energy."
"The reason for 802.11 to try and do therefore is to push fairness. If the sender fails to urge associate ACK for any fragment, it should abandon the transmission and re-contend for the medium in order that different nodes have an opportunity to transmit. This causes a protracted delay if the receiver very needs the whole message to begin process. In distinction, message passing lengthens the coordinated universal time and re-transmits this fragment. So it's fewer contentions and an occasional latency. There ought to be a limit on what percentage extensions are often created for each message just in case that the receiver is de facto out of order or lost in association throughout the transmission. However, for sensing element networks, application-level fairness is that the goal as hostile per-node fairness."

E. En Svgs vs accrued Latcy

Due to the respective nodes sleep schedules we can analyze the trade-offs betwn savings & latency. " One tends to compare S-MAC protocols with 802.11, which does have a sleep schedule. It may be noted that once a packet moves through a mlt-hp NW, it experiences delays at every hop and hence the advantage of S-MAC."

"Carrier sense delay happens once the sender performs carrier sense. Its worth is decided by the rivalry window size. Back off delay happens once carrier sense failing, either as a result of the node detects another transmission or collision happens. Transmission delay is clearly determined by channel information measure, packet length and therefore the secret writing or modulation theme adopted."

Propagation delay is depends on the gap between the causation and receiving nodes. In WSNs, node distance is often terribly tiny, and therefore the propagation delay will ordinarily be neglected.

Processing delay; the receiver has to method the packet before forwarding it to succeeding hop. This delay primarily depends on the computing power or OS of the node.
and therefore the potency of in-network processing algorithms.

Queuing delay is decided by the traffic. Within the serious traffic case, delay because of queuing becomes a dominant issue.

The preceding delays square measure inherent to a multi-hop network victimisation contention-based raincoat protocols. These factors square measure a similar for each S-MAC and 802.11 families of protocols. Periodic sleeping in S-MAC causes an additional delay. Once a sndr gts a pkt to xmit, it should wt till th rx wks up. Wve tend to decision it slp dly snc it's csed by th slp of th rx.

" In this case we tend to decision an entire cycle of the listen and sleep a frame. Assume a packet reaches the sender with equal chance in time at intervals a frame. Therefore "the average sleep delay on the sender is 

\[ D_s = \frac{T_{\text{frame}}}{2} \quad (1) \]

Where

\[ T_{\text{frame}} = T_{\text{listen}} + T_{\text{sleep}} \quad (2) \]

Comparing with protocols while not periodic sleep, the relative energy savings in S-MAC is "

\[ E_s = \frac{T_{\text{sleep}}}{T_{\text{frame}}} = 1 - \frac{T_{\text{listen}}}{T_{\text{frame}}} \quad (3) \]

"The last term within the higher than equation is that the duty cycle of the node. It's fascinating to own the listen time as short as attainable in order that for a definite duty cycle, the mean sleep delay is brief. Usually we tend to set the listen time as 300ms.

Figure 3.11 shows the share of energy savings atomic number 99 vs. mean sleep delay \( D_s \) on each node for the listen time of 300ms and 200ms. We are able to see that even though there's no sleep time or are zero (no sleeping) there's still a delay. This impact is as a result of rivalry solely starts at the start of every listen interval."
Since the primary goal of our work was not to compare S-MAC and 802.11 type protocols a test bed was not set up nor used. However interested readers are referred to http://www.cs.berkeley.edu/~awoo/smartdust/ and the work of Wei Ye, John Heidemann, and Deborah Estrin of UCB.

3.4. Low-Energy Adaptive Clustering Hierarchy (LEACH) Protocol

Traditionally, protocols and applications within the networking domain are designed to figure in large-scale heterogeneous and hierarchically organized networks with low failure rate during a Wireless sensing element.

In a Wireless Sensor Network (WSN) scenario, a new problem arises and traditional routing protocols cannot be successfully applied. Based on various findings it is realized that the conventional protocols of direct transmission then minimum transmission energy, multi-hop routing techniques, and static clustering may not be
optimal for sensor networks. LEACH (Low-Energy Adaptive Clustering Hierarchy), a clustering-based protocol utilizes randomized rotation of local cluster base stations (cluster-heads) to evenly distribute the energy load among the sensors in the network is a leading protocol in this class [241]. In this chapter we simulate the LEACH protocol for a sensor field and verify the energy savings achieved by LEACH.

3.4.1 Background: Evolution to LEACH

It can be seen how LEACH aids in monitoring a physical phenomenon by gathering and delivering information to the interested party efficiently. Sensor nodes are deployed into a particular field and perform certain tasks of sensing or tracking and convey information to a base station. The success of these operations can be attributed to the recent development of micro-electronic-mechanical systems (MEMS), processor, and radio and memory technologies.

"Eventually, the data being sensed by the nodes in the network must be transmitted to a control center or base station where the end-user can access the data and there are many possible models for these micro sensor networks. In this chapter, we consider micro sensor networks where:

- In this the BS is fxd & loc fr frm th snsrs.
- Nodes in the network are all homogeneous and energy constrained.

The primary task of any sensor network is met by efficient collaboration of sensors that means i) ensure connectivity ii) have an efficient role assignment iii) collect only significant data iv) decrease latency and v) save energy. Since wireless communications consume significant amounts of battery power and also sensor nodes should spend as little energy as possible receiving and transmitting data. It is
necessary for communication protocols to maximize nodes’ lifetimes.

Sensors networks need to cope with more frequent topological changes (not so much because of mobility but mainly because of nodes failing and going to sleep and being blocked by environment interference) and have as primary goal to prolong network lifetime by power conservation. Considerable research is focused in this area. One approach is that of designing localized algorithms where sensors only interact with other sensors in restricted vicinity and have at best an indirect global view or directed diffusion. Rumor Routing is a logical compromise between query and event flooding. With Rumor Routing paths (possibly multiple and non-optimal) are created leading to each event. Whenever a query is generated it is sent on a random walk until it crosses one of the paths leading to the event of interest.

In energy efficient protocols a 2-level hierarchical routing protocol (LEACH) is used which attempts to minimize global energy dissipation and distribute energy consumption evenly across all nodes and this is achieved by the formation of clusters with localized coordination and by rotating the high-energy cluster heads and by locally compressing data.

Another energy efficient protocol called PEGASIS (Power-Efficient Gathering in Sensor Information Systems), is near optimal for this data gathering application in sensor networks. The key idea in PEGASIS is to form a chain among the sensor nodes so that each node will receive from and transmit to a close neighbor. Gathered data moves from node to node-get fused and eventually a designated node transmits to the BS and nodes take turns transmitting to the BS so that the average energy spent by each node per round is reduced.
3.4.2 LEACH Protocol

Substantial energy gains can be achieved by performing in-network processing such as data fusion. Concretely, it is known that Wireless Sensor Networks (WSNs) contain too much or excess data that is not required by the end-user. Therefore automated methods of aggregation and data fusion can combine several unreliable data and reduce uncorrelated noise to form more accurate signals.

The authors of [242] developed Low-Energy Adaptive Clustering Hierarchy - (LEACH) a routing protocol for WSNs. This protocol was developed after extensive analysis of the advantages and disadvantages of conventional routing protocols.

The basic policy of LEACH is of a localized coordination and control. This implies that most communication and processing is done locally before transmitting the data to a central/base station. To understand the concept imagine a massively large WS Network deployed for a specific task(s) and controlled by a single base station. In such a scenario all the nodes would individually transmit their data to the base station. It is obvious that this direct transmission model is high on energy dissipation.

So how does LEACH conserve energy? It adopts a two stage strategy to achieve energy saving. In the first stage clusters are formed. For instance if there is a network of 100 nodes, then initially 10% or 10 nodes are randomly selected as cluster heads and ten clusters are formed. In the next stage members of clusters transmit their data to respective cluster heads. The cluster heads do in-network processing and then forward the fused data to the base station. This is the completion of one cycle. In the next cycle new cluster heads are selected forming new clusters.
large reduction in the energy dissipation, as computation is much cheaper than communication.

There are three models of communication in the context of WSNs. LEACH uses the clustering model. All the three models can be easily understood from the diagrams given below:

![Communication Models: (a) Direct Transmission (b) Multihop Transmission (c) Clustering](image)

**Figure 3.12:** Communication Models: (a) Direct Transmission (b) Multihop Transmission (c) Clustering. Mathematical models are given alongside.

The operation of LEACH is separated into two stages, the setup stage and the steady state stage. In the setup phase, the clusters are organized and CHs are selected. In the steady state stage, the actual data transfer to the base station takes place. In setup stage, each node chooses a random number between 0 and 1. If the number is less
It can thus be seen from the preceding description that the LEACH protocol follows a two step process - i) the set up phase and ii) the steady state phase. In i) the cluster heads are formed and clusters formed. Whilst in ii) the data collected by the CHs is processed in terms of data fusion and then transmitted to the central/base station. The completion of the two step process constitutes one cycle. Also as a cycle is completed the nodes assign themselves numeric values between 0 and 1.

In the next cycle new cluster heads are chosen based on the LEACH algorithm. The main criterion for choosing new cluster heads is that it should not already have been a cluster head, then a random selection is done on the basis of the numeric values of the nodes. It can thus be noted that the LEACH algorithm balances the high energy requirement of transmissions to the base station. This balancing is achieved by equitably loading all the nodes of the network as far as energy consumption is concerned. That is rotate cluster heads amongst all the nodes.

In other words, the LEACH algorithm implements a randomized rotation of the high-energy clusterhead position among the various sensors so as not to drain the battery of a single sensor. A local data fusion is also done to further reduce energy consumption. The basic data fusion process involves compressing the data at the cluster head position before transmitting it to the base station.

Another energy saving feature of LEACH is scheduling of transmissions within each cluster. This means that within the cluster a Time Division Multiplexing is done so that the nodes can have their sleep and listen cycles to conserve their energy.
Figure 3.13: (a) Initial random deployment of sensors for a task. (b) Cluster head selection. (c) Message passing amongst CHs and their nodes. (d) Data aggregation and fusion at CHs. (e) Transfer of information from CHs to the base station. The process of (a) to (e) is called one round in LEACH; the process is repeated by choosing next CHs.
Cluster Head Selection Algorithms

The sensor nodes select themselves as cluster heads according to a certain probability as outlined below:

- \( P_i(t) \) is the probability on which node \( i \) elects itself to be Cluster Head at the beginning of the round \( r+1 \) (which starts at time \( t \)) such that expected number of clusterhead nodes for this round is \( k \).

\[
E[\#CH] = \sum_{i=1}^{N} P_i(t) = k.
\]  

(4)

\( k \) is the number of clusters during each round

\( N \) is the number of nodes in the network

- Thus each node will be Cluster Head once in \( N/k \) rounds.
- And Probability for each node \( i \) to be a cluster-head at time \( t \):

\[
P_i(t) = \begin{cases} N - k \cdot \left( r \mod \left( \frac{N}{k} \right) \right) & C_i(t) = 1 \\ 0 & C_i(t) = 0 \end{cases}
\]  

(5)

\( C_i(t) \) is it determines whether node \( i \) has been a cluster head in most recent \( r \mod(N/k) \) rounds. Thus the expression:
\[ E \left[ \sum_{i=1}^{N} C_i(t) \right] = N - k \ast \left( r \mod \frac{N}{k} \right) \]

\[ \sum_{i=1}^{N} C_i(t) \]

= total no. of nodes eligible to be a **cluster-head** at time \( t \).

- This ensures energy at each node to be approx. equal after every \( N/k \) rounds.

- Using (2) and (3), expected # of Cluster Heads per round is,

\[
E[\#CH] = \sum_{i=1}^{N} P_i(t) \ast 1
\]

\[
= \left( N - k \ast \left( r \mod \frac{N}{k} \right) \right) \ast \frac{k}{N - k \ast \left( r \mod \frac{N}{k} \right)}
\]

\[ k \]

**Cluster Formation Algorithm**

- Cluster Heads sends an advertisement message (ADV) to all using CSMA MAC protocol.
  - **ADV** = node’s ID + distinguishable header.
- On the basis of the received signal value of ADV message, each non-Cluster Head node determines it’s Cluster Head for this round.
- Then, each non-Cluster Head transmits a join-request message (Join-REQ) back to its chosen Cluster Head using a CSMA MAC protocol.
Cluster Formation Algorithm

- Cluster Heads sends an advertisement message (ADV) to all using CSMA MAC protocol.
  - ADV = node’s ID + distinguishable header.

- On the basis of the received signal value of ADV message, its Cluster Head for that round is determined by non-CH nodes.

- Then, each non-Cluster Head transmits a join-request message (Join-REQ) back to its chosen Cluster Head using a CSMA MAC protocol.
  - Join-REQ = node’s ID + cluster-head ID + header.

- Cluster Head node schedules a TDMA sequence for data transmission coordination within the cluster.

- TDMA Schedule
  - Prevents collision among data messages.
  - Energy conservation in non cluster-head nodes.

A typical flow of the distributed cluster formation algorithm for LEACH [243] is shown in the flowchart below:
Figure 3.14: Flowchart of the distributed cluster formation algorithm for LEACH [243]

The time line for one round of LEACH is illustrated in [244]. To avoid collisions & increase sleep time of CHs, data transmissions are explicitly scheduled. This enables the non-cluster heads to remain in the sleep state for a longer time.

Figure 3.15: Time Line showing LEACH operation; Steady state phase.
Discussion

It can be seen that LEACH is a clustering-based routing protocol that minimizes global energy usage by distributing the load to all the nodes at various points in time. It outperforms other clustering algorithms by requiring nodes to volunteer to be high-energy cluster-heads and adapting the correlated clusters based on the nodes that choose to be cluster-heads. Excessive drain on the battery of a single sensor is not done as LEACH rotates the high energy cluster head, among all the sensors of a cluster/network.

A further energy saving feature is that even at the cluster head the data is compressed, reducing energy dissipation and hence increasing lifetime of the NW. We also note that within the steady state phase, the cluster head makes a TDM based scheduling for the members, so that notes are switched off when it is not their turn to transmit. The step-by-step flow of the LEACH algorithm is depicted in the diagram above, which is suitably annotated.

The LEACH protocol is completely distributed and decentralized in that it requires no control information from the base station or global network knowledge in order to operate. It is better than other algorithms on two counts: first, by random rotation of high energy cluster heads the dissipation is evenly distributed and second within the clusters, the sleep awake schedule further reduces the energy. We thus have a double effect so to say: In addition the common signal is compressed and made noise free also adds to reducing energy dissipation.

Analysis of the simulation results show that LEACH reduces energy consumption as much as 8 times as compared to direct and MTE methods of transmission. Also FND (first node dies) occurs 8 times larger than FND of MTE and direct transmission. The LND (last node death) is 3 minutes later than that of other protocols of the same
At this point of the work we note that large bodies of researchers are working on the communication stack of sensor networks. The efforts being directed are at refining routing protocols to increase network lifetime. The two main protocols analyzed have been a sensor specific direct transmission MAC protocol and clustering based LEACH protocol. The results show that LEACH is more energy efficient. The ground is thus set to explore the main hypothesis of our research that machine learning techniques further improve energy efficiency.

3.5 Simulation Framework & results

The simulation settings can be broadly categorized into two: new and adaptive developments. The latter developments are those simulation scenarios that existed before the idea of WSNs and obviously the new development are post-WSN emergence. The simulation tools of the research that are discussed in this section begin with an introduction (brief) of NS-2, followed by the model of NS-2 used in the work. Lastly is highlighted the simulation results that have been presented earlier in this chapter. A brief use of Simulink MATLAB has also been done and most importantly basic ML algorithm has been implemented using octave. The work in octave is well documented and attached as an appendix 1. Now the study begins in brief the two major simulation environments used in WSN namely NS2 and Matlab.

3.5.1 NS-2 Overview

NS, the network simulator is a simulation tool especially targeting research in the area of networking. Fig 3.16 depicts a typical networking architecture of mobile networking. NS2 includes support not only for different layers; other architectures but also the protocol. Support for energy constraints and node movements are contained
in the wireless model of NS. So by suitable modification this model can adapt to WSNs.

Fig. 3.16 — Foundation of the sensor network model used in NS-2

In [245,246] we find a detailed exposition of NS. The essential feature of this tool being that it is an object-oriented discrete event scenario creator. The object-oriented version of Tcl and C++ are used in NS2. In WSNs we use variation of the 802.11 families, Blue and IRUWB. Since these protocols are implemented in wireless, they can be implemented in WSN application also. The number of specific WSN protocols implemented in rather low despite a large body of researches work in the field. We need to take note of the work in [255] that studies a special WSN framework called “Mannasim”. This provides a stimulation framework for WSN specific protocols such as LEACH and Directed Diffusion.

An extension of this framework, primarily for intrusion detection is found in NS2 – MUIN [247] and has features required to model a sensor network namely, power (battery and radio), channel model, and scenario generation. These models are provided by a widely used simulation framework Sensorsim [248-250]. In [251] the author highlights the flexibilities of the framework. The flexibility to investigate the
characteristics of various sensor networks is incorporated as it already contains flexible models for ad-hoc wireless networks. In this case the NS2 scenario are built with many of the same sets of protocols that are available in real world. Figure 3.17 shows the support for different protocols in the network layers. A detailed description of these aspects is given in [252 and 253].

Figure 3.17: Support for different layers in NS-2

3.5.1 The Extended ns-2 Architecture

Sensor Network Extension

It was mentioned earlier that NS2 is based on C++ and OTcl. The OTcl script does the following:

Initiates an even scheduler; sets up the network topology using the network object and tells the traffic sources when to start/stop transmitting packets. Simply put, the general architecture of NS2 following the TCP/IP protocols stack.

To set up a sensor network in NS2 one follows the same format as that of mobile mote stimulations. To create your own stimulation it is best to modify the code of the example.

Briefly the steps can be enumerated as below:

1. Configure phenomenon/data channels. Firstly all phenomenon modes should be configured on the same channel, even if they are emanating
different types of phenomenon (temperature, pressure, etc.). Sensors mode having data should be on a different channel to avoid contention of media.

2. MAC protocol. In this case phenomenon would eliminate emanate without regards to collisions. An appropriate MAC layer protocol from 802.11 families has to be chosen. This has to be done for the phenomenon channel. This is invoked by two commands; set val (mac) and set val (PHENOM mac).

3. Configure phenomenon nodes. This is done using node-config, but PHENOM is specified as the routing protocol. So phenomenon emanate as per this protocol. Sample node configuration statements can be found.

4. Pulse rate and type. These parameters of a mode are configured using the commands:

pulse rate FLOAT and ‘phenomenon pattern’. These commands enable the broadcast frequency or rate of the phenomenon node as also the type of sensing mode sound, seismic, etc.

5. The Sensor Motes. These must be with the PHENOM channel attribute and the – channel attribute. Importantly the energy model containing transmit, receive power, etc. need to be defined.

6. Data Collection Points. These are specific commands for particular motes and the gateway node (one sending to the base station).

7. Attach Sensor Agents; UDP Agents. These define how sensor nodes react once the target or phenomenon is detected. The foregoing clearly highlights the flexibility of NS2 for different topologies and configurations.
The foregoing clearly highlights the flexibility of NS2 for different topologies and configurations

**Application specific NS-2.**

The object and function of ns-2 have to be configured and implemented depending on the phenomenon. So the phenomenon notes (or sensing field) trigger sensor nodes and traffic is generated. This traffic depends on the function the SN is performing. For instance the network function is to track moving object (a vehicle) the traffic would be more than of a design to provide an outside observer with raw data. The files that implement the changes are: phenom/phenom.cc, h; sensornets-NRL/ sensoragent.cc, h, sensornets-NRL/ phenom_packet.h. The manual explains the detail of these fields and methods to implement each file for a specific application. Figure 3.18 depicts a typical scenario.

**Changes to ns2 extension**

In figure 3.19 is highlighted the manner in which changes have been done to the ns2 framework. Basically these are modifications and additions. The blocks on the left indicate the objects that were modified and on the right hand side are the ones that were added for instance “tcl/lib/ns-lib.tcl”; this component is infrastructure related and interprets node configuration. In a sensor nature the extension introduce two new mote types- the sensor node and phenomenon node. So some arguments are added in the node-config function to adapt them. Similarly the other component is modified. (Details of modifications are at types)
Fig. 3.18: Schematic of twenty five sensor nodes randomly deployed

-BLANK SPACE LEFT INTENTIONALLY TO FIT NEXT DIAGRAM-
3.5.3 WSN with MATLAB

The MATLAB environment has been extensively used by researches to validate WSN algorithms and protocols. Detailed implementation of a WSN using Matlab is contained as an Appendix by the relevant title. It will show the steps to simulate the
lifetime of WSN. Using this basic methodology is carried out various simulations. A simple model using the simulation guideline is at fig 3.20. It may be noted that the following architecture of the system need to be configured i) the transmitter ii) the medium access iii) receiver

![Figure 3.20. A Simple WSN Model](image)

### 3.6 Results and Analysis.

The foregoing sections have highlighted the two main and popular simulation environments of NS-2 and Matlab. The examples given are basic ones to enable us to grasp the semantics and functioning of these two environments. By understanding the foundations we are apply to specifically apply to the two routing protocols that have been discussed in this chapter.

Needless to say that sound knowledge and understanding of ns-2 and Matlab is by itself no small achievement. The first output generated is that of a random deployment of 100 sensors in a designated area performing a typical environment monitoring task. It should also be noted that the results are based on the default leach
test set by MIT. Initial results use the default.config file. Then the parameters are changed to get different results.

![Random Deployment of 100 nodes with defined parameters](image)

<table>
<thead>
<tr>
<th>Nodes</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network size</td>
<td>100 m × 100 m</td>
</tr>
<tr>
<td>Base station location</td>
<td>(50, 175)</td>
</tr>
<tr>
<td>Radio propagation speed</td>
<td>$3 \times 10^8$ m/s</td>
</tr>
<tr>
<td>Processing delay</td>
<td>50 µs</td>
</tr>
<tr>
<td>Radio speed</td>
<td>1 Mbps</td>
</tr>
<tr>
<td>Data size</td>
<td>500 bytes</td>
</tr>
</tbody>
</table>

**Figure 3.21:** Random Deployment of 100 nodes with defined parameters
Figure 3.22: LEACH distributes more data per unit energy than minimum transmission-energy (MTE) routing protocol. (a) The number of data signals received at the base station as time lapses becomes constant but in the case of LEACH it increases. (b) LEACH delivers 40% more data per unit energy than MTE.

Figure 3.23: Optimum No Clusters versus energy dissipation
Figure 3.24: System lifetime using MTE routing and LEACH with 0.5 J/node. (a) No of nodes alive versus time; note 100 nodes are alive longer than for MTE. (b) Hence LEACH gives more No of data items received at the BS.
Figure 3.25 Dynamic clusters: (a) cluster-head nodes = C at time $t_1$ (b) cluster-head nodes = C at time $t_1 + d$ (All nodes marked with a given symbol belong to the same cluster, and the clusterhead nodes are marked with a red/black circle.)

Summary

An important routing protocol LEACH is based on cluster formation. The protocol minimizes global energy consumption by distributing the load to nodes at various times. On the other hand cluster-heads are designated as such for the entire time period. So LEACH outperform these algorithms because all nodes become CHs at different times and also within a cluster many nodes are in “sleep” states as they are not transmitting all the time, but only during their “slotted” time. The
LEACH algorithm is completely distributed, therefore a knowledge of the global network by the nodes is not required. The only drawback being that it is not ideal for large geographical areas.

Figure 3.26: Machine Learning (a) Training Examples (b) Decision Boundary.
3.7 Machine Learning Results

The subsequent diagrams illustrate regularized regression variance and bias.

```matlab
plot(X, y, 'rx', 'MarkerSize', 10, 'LineWidth', 1.5);
```
== Connecting to ml-class ...
== [ml-class] Submitted Assignment 5 - Part 1 - Regularized Linear Regression Cost Function
== Nice work!

== [ml-class] Submitted Assignment 5 - Part 2 - Regularized Linear Regression Gradient
== Nice work!
Iteration 14 | Cost: 2.436413e-33
Iteration 13 | Cost: 1.655087e-01
Iteration 11 | Cost: 1.608881e-01
Iteration 10 | Cost: 2.461099e-01
Iteration 11 | Cost: 2.012938e-01
Iteration 7  | Cost: 1.752179e-01
Iteration 15 | Cost: 1.538717e-01
Iteration 8  | Cost: 1.360020e-01
Iteration 15 | Cost: 1.253141e-01
Iteration 13 | Cost: 1.155151e-01

== [ml-class] Submitted Assignment 5 - Part 3 - Learning Curve
== Nice work!

== [ml-class] Submitted Assignment 5 - Part 4 - Polynomial Feature Mapping
== Nice work!
Iteration 16 | Cost: 1.151643e-01
Iteration 7  | Cost: 1.151647e-01
3.8 Conclusion

This chapter is perhaps the most important of the thesis as it pertains to the main results of the work. It has combined and evaluated the important energy saving techniques. Starting from the basic MTE technique to LEACH it was finally seen that machine learning techniques further enhance the network lifetime. In the investigations and analysis it was clearly seen that as far as sensor networks are concerned specialized protocols are needed due to the AdHoc nature of sensor networks. The study analyzed important work in this field namely LEACH and further studied the implementation of machine learning techniques to fulfill the thesis goals.