Journal of
Sensor and Actuator Networks

MDPI

Article

Energy-Aware Wireless Sensor Networks for Smart Buildings: A Review

Najem Naji [1,2,*], Mohamed Riduan Abid [2], Nissrine Krami [1] and Driss Benhaddou [3]

[1] École Nationale des Sciences Appliquées, Ibn-Tofail University, Kenitra 14000, Morocco; krami.nissrine@univ-ibntofail.ac.ma
[2] School of Science and Engineering, Al Akhawayn University in Ifrane, Ifrane 53000, Morocco; r.abid@aui.ma
[3] Engineering Technology Department, University of Houston, Houston, TX 77204, USA; dbenhaddou@uh.edu
* Correspondence: n.naji@aui.ma

Abstract: The design of Wireless Sensor Networks (WSN) requires the fulfillment of several design requirements. The most important one is optimizing the battery's lifetime, which is tightly coupled to the sensor lifetime. End-users usually avoid replacing sensors' batteries, especially in massive deployment scenarios like smart agriculture and smart buildings. To optimize battery lifetime, wireless sensor designers need to delineate and optimize active components at different levels of the sensor's layered architecture, mainly, (1) the number of data sets being generated and processed at the application layer, (2) the size and the architecture of the operating systems (OS), (3) the networking layers' protocols, and (4) the architecture of electronic components and duty cycling techniques. This paper reviews the different relevant technologies and investigates how they optimize energy consumption at each layer of the sensor's architecture, e.g., hardware, operating system, application, and networking layer. This paper aims to make the researcher aware of the various optimization opportunities when designing WSN nodes. To our knowledge, there is no other work in the literature that reviews energy optimization of WSN in the context of Smart Energy-Efficient Buildings (SEEB) and from the formerly four listed perspectives to help in the design and implementation of optimal WSN for SEEB.

Keywords: Wireless Sensor Network; IoT; smart buildings; energy efficiency; wireless mesh networks; operating systems

check for updates

Citation: Naji, N.; Abid, M.R.; Krami, N.; Benhaddou, D. Energy-Aware Wireless Sensor Networks for Smart Buildings: A Review. *J. Sens. Actuator Netw.* **2021**, *10*, 67. https://doi.org/10.3390/jsan10040067

Academic Editors: Subhas Mukhopadhyay, Yoshiyasu Takefuji and Enrico Vezzetti

Received: 22 September 2021
Accepted: 19 November 2021
Published: 26 November 2021

1. Introduction

Wireless Sensor Network suffers from limited battery size, and sensor nodes are sometimes deployed in an environment where a person cannot replace sensor batteries frequently. Therefore, energy efficiency is a critical design factor in sensor nodes. In WSNs, energy efficiency is achieved through the optimal design of sensor and actuator nodes. The optimal design tends to provide efficient energy supply mechanisms and then efficient energy consumption. Moreover, the sensor design should take into consideration the deployment environment and the available energy source. These design requirements are because energy is a critical resource for a WSN node.

In IoT devices, the data rate, programming model, and processes at the application layer affect energy consumption, which manages data acquisition and processing. The operating system's energy consumption depends on the system processes and services that are running. At the networking layer, routing protocols and data link protocols affect the energy consumption as they join the network, discover routes, and transmit data. The hardware platform's energy consumption relies on the complexity of the architecture and electronic components. Finally, the source of the power, such as the battery, tends to provide maximum capacity while the size of the battery is small. Researchers [1] surveyed Energy Harvesting (EH) techniques such as materials, technology, photovoltaic, piezoelectric converters, etc. EH techniques also include hybridization that tends to enable the autonomy of power supply in IoT devices.

This paper tends to make the researcher aware of the energy optimization techniques and opportunities in IoT and WSN for Smart Energy-Efficient Buildings. Therefore, the paper presents different approaches, methods, and technologies based on the layered architecture of the IoT device. Hence, it guides the researcher to design energy-aware IoT devices for Smart Energy-Efficient Buildings.

The rest of the paper is organized as follows. Section 2 presents a literature review of WSN design challenges. Section 3 discusses the hardware resources and presents different examples that can be used. The different operating systems and embedded systems' alternatives are delineated in Section 4, and we list the various data acquisition techniques and their impact on energy consumption optimization. In Section 5, we list data acquisition techniques at the application layer. Then, in Section 6, we emphasize the network and data link protocols that are energy-aware, along with a comparative study of relevant network topologies. In Section 7, we present different battery types that can be used to power the WSN nodes. Last but not least, we conclude by presenting a framework recommending optimal components to adopt when designing a static WSN for smart buildings.

2. Literature Review

In Wireless Sensor Networks (WSN), sensor nodes are deployed either uniformly (controlled), randomly (uncontrolled), or in a hybrid manner [2]. In uniform deployment, sensors are placed based on pre-calculated and pre-engineered network architecture, while nodes are deployed at random positions in the random deployment. For uniform deployment, network connectivity and coverage are controlled. The random deployment is used in large WSNs where sensor nodes' connectivity and coverage are subject to random network modeling. The deployment choice depends mainly on the accessibility to the deployment environment.

On the one hand, deploying sensor nodes in a harsh environment like a battlefield, disaster region, or a forest can be possible only by throwing sensors from an airplane. Hence, the deployment, in such a case, will be random. On the other hand, uniform deployment is the placement of sensor nodes in a controlled manner. This type of deployment is used in a user-friendly environment such as a Smart Building. Since the Smart Building is our deployment environment, in this paper, we discuss the design criteria and challenges of indoor WSN.

Several challenges arise while designing a WSN [3], such as network connectivity, network coverage, data fidelity, and network lifetime [4]. In the context of a Smart Building, network coverage, data fidelity, and network connectivity issues can be handled by the uniform deployment (Figure 1). Good coverage and connectivity can be achieved by well positioning nodes and choosing devices with the optimal signal range. Additionally, data fidelity can be ensured by placing multiple sensors to sample a single area. For example, to measure temperature value in a room, there might be a need to place numerous temperature sensors where the average of the sampled values represents the real temperature value of the room. However, in all WSN types (indoor, outdoor, static, or even mobile) [5], extending the network's lifetime to the maximum is a goal that is needed whenever there is room for doing so. This challenge can be achieved by optimizing the energy consumption at the level of WSN nodes.

Based on the literature review of WSN, this paper examines the network lifetime with the focus on the energy consumption of the sensor node at the hardware layer, operating system layer, networking layer, and application layer (see Figure 2). This study took into consideration three factors: first, the network topologies such as Mesh, Cluster, Tree, and Star topologies; second, the node's role in the WSN such as End node, Router node, Cluster-Head, or Gateway; third, the investigations included the data processing approaches where WSN nodes are involved, for instance, In-node processing and In-network processing. As a result of this study, this paper guides the WSN designer on extending the WSN's lifetime by optimizing the node's resources.

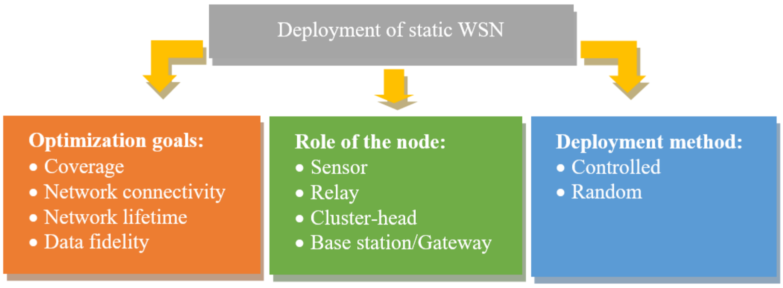

Figure 1. WSN deployment criteria.

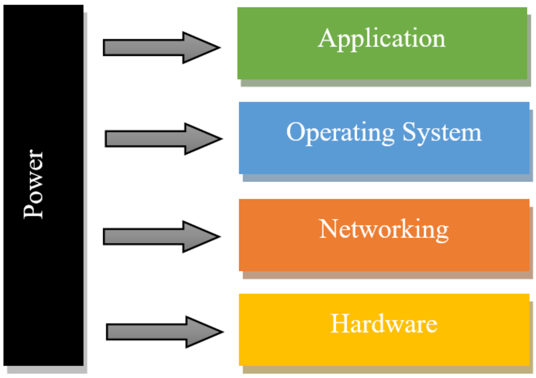

Figure 2. WSN node Architecture's layers.

3. Hardware Layer

In general, the hardware platform of a sensor node contains four components: the hardware platform, the communication unit, the sensor module, and the battery [6]. In this section, we present both the hardware platform and the communication unit. For the battery component, we will dedicate a whole section later in this paper.

3.1. The Hardware Platform (Microcontroller and CPU)

The hardware platforms (boards) are of two types: CPU-based and microcontroller-based [7]. First, the CPU-based platforms contain the following elements: CPU, RAM, ROM, Timer, and I/O ports, while Microcontroller-based hardware platforms include the same components as the CPU-based ones but on a single ship [8]. The Analog-to-Digital Converter (A/D) [9] receives the signal from the sensing module, converts it, and then transmits it to the processing unit, which can be either part of the hardware platform or part of the sensing module itself. Currently, most platforms used in a WSN are microcontroller based.

Energy optimization at the hardware level can be achieved through the computer system architecture and the CPU architecture. In the computer system architecture, Harvard architecture [10] is a computer architecture that physically separates storage and signal pathways for instructions and data. In such architecture, instructions of a program can be stored in read-only memory, while data memory generally requires read-write memory. Because there might be more instruction memory than data memory, instruction addresses are wider than data addresses. Harvard has taken an application-driven approach by including hardware acceleration to optimize typical operations in WSN applications [11]. The Harvard event-driven system for WSN uses three techniques to reduce energy consumption [12].

Lightweight event handling in hardware: For handling incoming interrupts, a specialized Event Processor is given responsibility for doing so. As a result, it removes the software overhead that provides event handling on a general-purpose processor.

Hardware acceleration for typical WSN tasks: Modular hardware accelerators are included to complete regular application tasks such as data filtering and message routing.

Application-controlled fine-grained VDD-gating: Addressing leakage current with architecture support for VDD-gating enables accelerator blocks to be powered off when unused.

The most used CPU architecture in WSN nodes is the Reduced Instruction Set Computer architecture (RISC) [13]. This architecture is the one used in ARM CPUs that are energy efficient [14]. The RISC is an instruction set architecture (ISA) [15] that has a set of attributes that allows it to have lower cycles per instruction (CPI) than a complex instruction set computer (CISC). Most RISC architectures have fixed-length instructions (commonly 32 bits) and a simple encoding. One drawback of 32-bit instructions is reduced code density, which is more adverse, characteristic in embedded systems where RISC architectures were originally designed to serve. The architecture incorporates fewer clocks per Instruction. In contrast, the CISC processors use a higher clock rate to speed up processing, and that is why RISC architecture leads to consuming less energy. Here we present three types of microcontrollers [16].

First, PIC microcontrollers (Peripheral Interface Controller) [17] can operate over a wide frequency and voltage range, making frequency scaling possible. PIC16 microcontrollers have only one sleep mode, which was placed on the processor core. The PIC18 series have two sleep modes. The core is shut off if the peripherals are running, and vice versa for the second sleep mode. Second, The AVR microcontroller [18] follows the Harvard architecture. ATMega128 microcontroller, which belongs to the AVR family, is widely used in sensor nodes. CPU, Flash, EEPROM, and SRAM are single integrated chips. It has six power modes: Idle, ADC noise reduction, power-save, power-down, standby, and extended standby. Third, the MSP430 Microcontroller can be used for low-powered embedded devices [19]. The MSP430 also uses six different low-power modes, which can disable unneeded clocks and CPU. Table 1 illustrates the most used Microcontroller in WSN nodes.

Table 1. Microcontrollers used in sensor nodes.

Features Microcontroller	Architecture	CPU	RAM	Storage	Input Voltage	Power Consumption	Wake-Up Time
MSP430F5529	Mixed (von-Neumann and RISC	25 Mhz	10 KB	128 Kbytes	3.6 v	Active: 404 µA/1 MHz Standby: 2.5 µA/1 MHz	3.5 ns
MSP430F5514	Mixed (von-Neumann and RISC	25 Mhz	6	64	3.6	Active: 404 µA/1 MHz Standby: 2.5 µA/1 MHz	3.5 ns
Atmega128	AVR (modified Harvard)	16 Mhz	4 Kbytes	128 Kbytes	5.5 v	Active: 19 mA/1 MHz Standby: 11 mA/1 MHz Power save mode: 15 µA/1 MHz	65 ms
Atmega328	AVR (modified Harvard)	16 Mhz	2 Kbytes	32 Kbytes	5.5 v	Active: 0.2 mA/1 MHz Standby: 0.1 µA/1 MHz Power save mode: 0.75 µA/1MHz	65 ms
PIC16F877A	Harvard	20 Mhz	368 Bytes	8 Kbytes	5.5 v	Active: 15 mA/1 MHz Standby: 7 nA/1 MHz Power save mode: 0.75 µA/1 MHz	72 ms
PIC18F46J50	Harvard	48 Mhz	16 Kbytes	4 Kbytes	5.5 v	Active: 19 µA/1 MHz Standby: 1.5 µA/1 MHz Power save mode: 0.75 µA/1 MHz	105 ns

The input voltage provided in the table above is the maximum possible for each Microcontroller. As the Microcontroller can operate with lower voltage input, the CPU frequency is a function of the input voltage.

3.2. Radio Frequency Module

The communication unit consists of a wireless communication channel where Radio Frequency communication is based on electromagnetic waves. The radio link between two communication units specifies the modulation scheme, the frequency spectrum, and the RF power [20]. Here we present the most used RF module, which is the XBee device, then we refer to the energy-aware modulation techniques.

Nowadays, XBee RF modules from Digi are widely used in WSN. They were designed to meet IEEE 802.15.4 [21] standards to support the need for energy-efficient sensors. The module operates within the ISM 2.4-GHz frequency band. Table 2 illustrates the key features of the XBee and XBee Pro RF modules [22]:

Table 2. XBee and XBee Pro features.

XBee RF Features	XBee	XBee Pro
Indoor	Up to 30 m	Up to 90 m
Transmit power	1 mW	63 mW
Data rate	250 Kps	250 Kps
TX current	45 mA	250 mA
RX current	50 mA	55 mA
Power down current	<10 µA	<10 µA
ADC converter	Yes	Yes
Input voltage	3.3 V	3.3 V

These RF modules have five modes of operation: command mode, transmit mode, receive mode, sleep mode, and idle mode [23]. When the module is set to Idle mode, it does not send or receive data. The module shifts to the transmit mode when the data are received in the DI Buffer, and it shifts to receive mode when valid RF data are received through the antenna. The device goes to sleep mode when the sleep mode condition is met. The command mode is the mode through which the communication with the module is done to configure it. Since the scope of this paper is energy efficiency in IoT devices, we only focused on the advantages of the sleep mode and how it works to reduce energy consumption.

Two conditions must be met to enable the XBee to enter the sleep mode: First, pin 9 must be asserted while the module is in a pin sleep mode (where the SM parameter equals 1, 2, or 5). Second, the module is idle for a preconfigured amount of time. This amount of time can be defined by the ST (Time before Sleep) parameter during the configuration of the XBee RF module. Table 3 illustrates the power consumed during the different sleep modes of the XBee.

Table 3. Wake-up time and power consumption in sleep modes.

Sleep Mode	Wake-Up Time	Power Consumption at VCC 5 V
Pin Hebirnate (SM = 1)	13.2 ms	<10 µA
Pin Doze (SM = 2)	2 ms	<50 µA
Cyclic Sleep (SM = 4)	2 ms	<50 µA
Cyclic Sleep with pin wake-up (SM = 5)	2 ms	<50 µA

Pin hibernate: This mode is voltage level-activated; when pin9 is asserted, the module will finish any transmit, receive, or association activities, enter Idle Mode, and then enter a state of sleep. In such a case, the module will not respond to either serial or RF activity. To wake it up, pin 9 must be de-asserted, and it will be ready to send and receive data only when the CTS line is low. When the module is set to transmission mode, pin 9 should be de-asserted at least twice to guarantee enough time for data to enter the DI buffer and then be sent. Another wake-up mode is Pin Doze, which is faster but with higher power consumption.

The Cyclic Sleep Mode allows modules to listen to any incoming RF data. When the SM parameter is set to '4', the module is put to sleep; then, it wakes up once a cycle to receive any data coming from a Cyclic Sleep Coordinator. The module sends a pull request to the coordinator at a specific interval configured by the SP (Cyclic Sleep Period) parameter. The CTS will go low each time the module wakes up. The coordinator will transmit any queued data addressed to that module. If no data are put in the queue for that module, the coordinator does not transmit any data. At this time, the module will go to sleep for another cycle. If queued data are sent back to the module, it will stay awake for a while to allow for back and forth communication until the ST (Time before Sleep) timer expires. The cyclic sleep coordinator is always awake (SM = 6).

Cyclic Sleep Remote mode (Pin Wake-up SM = 5): This mode wakes up a remote sleeping module through RF interface or Pin9 de-assertion. It works as the previous mode with the addition of a pin-controlled wake-up at the remote module. The module will wake when a low is detected and then sets CTS low when ready to transmit or receive data. Any activity will reset the ST so that the module will go back to sleep when no signal is detected before the expiration of the timer. The module goes back to sleep based on the ST time, not the state of the pin.

3.3. Energy-Aware Modulation Techniques

Several aspects affect a radio module's power consumption, such as the modulation technique, data rate, and transmission power. Starting from the next paragraph, we focus on the modulation techniques used in energy-efficient WSN.

The modulation is the process of coding information into the carrier frequency [24]. This process includes Amplitude Shift Keying, Frequency Shift Keying, or Phase Shift Keying. The choice of the modulation technique strongly impacts energy consumption. At the MAC layer, various multi-access methods have been evaluated regarding energy cost (we will see them later in this paper). In [25–37], Modulation techniques have been investigated in terms of energy efficiency from the physical layer perspective. In these investigations, the RF transceiver is often burst mode; it is shut off most of the time with a duty cycle of no more than 1%. Modeling the transceiver power consumption can help reduce the energy consumed, but an inaccurate model will lead to an incorrect result. For example, if a model assumes that the transceiver power consumption is proportional to the RF transmit power, which scales with the data rate, the transmit power must be increased to maintain the same transmission distance. Hence, to increase the battery life of the sensor, the data rate then must be reduced. However, the transmitter power consumption is not related to the data rate due to the fixed power cost of the transmitter.

The electronic circuit components dominate the power consumption for RF transceivers that operate in the Gigahertz carrier frequency range. As a result, it is important first to model the circuit power consumption and then to model the communication protocols, modulation techniques, data rate, and channel conditions. This interplay between the communication protocols and the transceiver electronics is complicated and has rarely been investigated. One of the energy-aware modulation techniques is On/Off Key (OOK) modulation [38]. OOK is a special case of ASK modulation as no carrier is present when transmitting zero bits. Due to the simplicity and low cost of modulation, OOK puts the transmitter to idle mode when sending zero bits, which helps in reducing power consumption. Table 4 illustrates the overall system performance of the most known modulation techniques used in WSN.

Table 4. Energy-efficient modulation techniques.

Features Modulation Technique	System Performance
BPSK, QPSK, 16QAM, 64QAM	Good
BPSK, QPSK, 16QAM, 64QAM	Efficient physical layer design
BPSK, QPSK, 16QAM, 64QAM	Better energy efficiency
QPSK, 16QAM	BCH code is comparatively better than other codes
MPSK MQAM MFSK	Suitable for short-distance communication
MPSK MQAM MFSK	Analog decodes are energy efficient

Based on the review of the different modulation techniques, the design of a wake-up radio module is a real challenge for the communication block unit. Hence, turning off the RF module in sensors is highly recommended because it is usually more power-consuming than other components. However, this technique will lead to missing events as the sensor can no longer receive data. The challenge is to develop a low-power communication channel to wake up neighboring nodes when needed, which is still an area of active research in chip design, modulation, and RF communication techniques.

4. Operating System Layer

Due to the predefined purpose of nodes in WSN, the operating system can be sharpened to the needed features only. As a result, WSN nodes use lightweight operating systems such as TinyOS, Contiki, MANTIS, Nano-RK, LiteOS [39], etc. In some other cases, WSN nodes do not run an operating system, as we said before, but, instead, they run an embedded code (boot loader). The boot loader receives the user application in an assembly code format and runs it as a routine for a predefined amount of time, such as the boot loader in an Arduino device [40]. This section examines the key features of the famous operating systems dedicated to WSN nodes. Additionally, it recommends when to use which, based on the node's role, the network's topology, and the processing approach (Table 5).

Table 5. IoT operating systems' features.

Operating System Features	TinyOS	Contiki	Free RTOS	MANTIS	Nano-RK	LiteOS
Architecture	Monolithic	Modular	Microkernel RTOS	Virtual machine	Monolithic	Modular
Supported microcontrollers	AVR MSP430	AVR MSP430 ARM PIC32 Cortex-M	AVR MSP430 ARM	AVR MSP430	AVR MSP430	AVR
Programming model	Event-driven	Event-driven Multi-threading	Multi-threading	Event-driven Multi-threading	Multi-threading	Event-driven Multi-threading
License	BSD	BSD	GPL	BSD	GPL	GPL

4.1. TinyOS

TinyOS is an open-source and application-specific operating system designed for WSN fit in 400 bytes [41]. It is a multi-task non-preemptive operating system that uses the First In, First Out (FIFO) scheduling algorithm and libraries for network protocols, sensor drivers, and data acquisition tools.

4.2. Contiki

Contiki is a lightweight, open-source OS designed for WSN sensor nodes [42]. Contiki is a highly portable OS that is built around an event-driven kernel. Contiki provides a multitasking kernel, preemptive multithreading, TCP/IP networking, IPv6, a Graphical User Interface, a web browser, a personal web server, a simple telnet client, a screensaver, and virtual network computing.

4.3. Mantis

The MultimodAl system for Networks of In-situ Sensors (MANTIS) provides a new, multithreaded operating system for WSN that supports remote management of sensor nodes [43]. It is a lightweight, portable, and energy-efficient operating system that requires only 500 bytes. This memory includes kernel, scheduler, and network stack. Therefore, the applications can be tested on a PDA or a PC before moving it to the sensor node.

4.4. Nano-RK

Nano-RK is a preemptive, multitasking, real-time OS for WSN that is written in C [44]. It provides priority-based scheduling, timeliness, and WSN lifetime. Additionally, it allows limiting the application resource usage and network bandwidth reservations. For example, a task might only be allowed to execute 5 ms every 50 ms (CPU reservation) or a node might only be allowed to transmit one network packet per minute (network reservation). These reservations help to extend the battery lifetime.

4.5. LiteOS

LiteOS is a Unix-like operating system designed for WSN to provide programmers with a familiar programming paradigm [45]. It provides a thread-based programming mode, supports registering to event handlers using callbacks, provides a hierarchical file system, and supports object-oriented programming in the form of LiteC++ and a Unix-like shell. It only requires an 8-MHz CPU, 128 bytes of program flash, and 4 Kbytes of RAM.

5. Application Layer

In WSN, the sensors periodically sense incoming data, receive it, process it, and transmit it to the next node or the base station. In each application, the number of sensors and the frequency of data sampling are specific. In data acquisition, one of the main constraints is energy optimization.

5.1. Directed Diffusion

In Directed Diffusion technique, the data collection process involves data messages, gradients, and reinforcements [46]. An interesting message is a query where the user specifies their interest in specific data sets. The interest carries a sensing task where data describes the environment that is being sensed. In this data collection technique, data are named using attribute-value pairs. The sensing request is disseminated throughout the WSN as an interest for named data and sets up gradients within the WSN to draw events. The gradient is a direction state created in each node that receives an interest. The direction is set towards the neighboring node that will send the interest. The WSN events flow towards the originators of the interests through different paths where WSN reinforces one or a small number of these paths.

5.2. Chain Construction Protocol

In Chain-based protocol, the technique consists of constructing a transmission chain that connects all nodes to optimize energy consumption in each round of data collection [47]. A linear chain connects all the sensor nodes within the network. In this technique, data are transmitted from one end of the chain to the other end, where each node appends data to the received message in order to construct a larger packet, then forwards it to the next node. To optimize energy consumption, sensor nodes of the network keep the same size of the packet header. Hence, WSN nodes transmit the same number of bits except for the end nodes, as they only transmit their data. Using this technique of distributing the energy usage among nodes, the WSN prevents nodes from being overused. When a WSN node dies, the transmitter node skips the dead node and transmits data to the next living neighbor in the chain.

5.3. Probabilistic Model

This model relies on replicated dynamic probabilistic models to minimize transmissions between WSN nodes and the base station [48]. In this model, the WSN maintains a pair of dynamic probabilistic models. One copy is distributed over the WSN nodes, and the other is saved at the base station. Each time, the base station computes the expected values sampled by the WSN nodes according to the model that does not require WSN nodes' data. When WSN nodes sense data that the model cannot predict, they forward the data towards the base station, that is, information is communicated to the base station only when the base station's predictions are not valid.

5.4. A Simple, Single Cluster Model Using Data Correlation

This model is based on data correlation exploitation [49]. In this model, the WSN nodes that belong to the same cluster have a strong internal correlation. Therefore, the model takes advantage of this correlation to analyze the mutual effect of cluster size and the distance from the base station on reducing total WSN energy consumption. It is difficult to find clusters with optimal size; however, this model tries to obtain a near-optimal solution to construct energy-efficient clusters in the WSN network. In this model, heterogeneous-sized clusters demonstrated better energy efficiency in WSN.

5.5. Query Model

Before talking about this model, the two most important parts of data communication must be stated: query processing and data aggregation [50]. In addition to energy conservation, this model only provides useful data sets. Query approaches are based on a query agent that provides APIs that map user queries to lower-level semantics corresponding to the underlying routing and aggregating techniques. Query-based approaches support unicast, multicast, and broadcast communication models. Query agents will support a wide variety of routing and aggregation protocols, selecting the best combination based on the query type.

5.6. Data Aggregation Techniques

Data aggregation is a process of aggregating the sensor data using aggregation approaches. The algorithm uses the sensor data from the sensor node. It then aggregates the data by using some aggregation algorithms such as the centralized approach, LEACH (low-energy adaptive clustering hierarchy) [51], TAG (Tiny Aggregation) [52], etc. These aggregated data are transferred to the sink node by selecting the efficient path.

Centralized Approach: This is an address-centric approach where each node sends data to a central node via the shortest possible route using a multi-hop wireless protocol [53]. The sensor nodes simply send the data packets to a leader, which is the powerful node. The leader aggregates the data, which can be queried. Each intermediate node has to send the data packets addressed to the leader from the child nodes. In such a scenario, many messages have to be transmitted where the total number of messages equals the sum of external path lengths for each node.

In-Network Aggregation: In-network aggregation is the global process of gathering and routing information through a multi-hop network. It processes data at intermediate nodes to reduce energy consumption, hence, increasing network lifetime [54]. In In-Network aggregation, there are two approaches. First, with size reduction, data sets are combined and compressed to reduce the data packet's length. In the second, with merging data packets, received data from diverse neighbors are merged into one data packet to be transmitted to the sink.

Tree-Based approach: This approach is the construction of an aggregation tree with a minimum spanning tree. In this tree, data are forwarded to the sink where data sources are the WSN nodes, considered as leaves [55]. In this tree, each node relays data to its parent. The data flow in the tree starts from leaves (nodes) up to the sink where parent nodes do the data aggregation.

Cluster-Based Approach: In this approach, the WSN network is divided into clusters [56]. In each cluster, a cluster head is elected to aggregate data from WSN nodes of the cluster, and cluster heads establish a network to route the aggregated data to the sink.

COUGAR Approach: This approach suggests a query layer that supports data aggregation based on queries [57]. In this approach, users can issue queries to the WSN where the query layer processes user queries and generates a cost-effective query plan. The main parameter that is considered by this approach is the communication cost, which includes the cost of routing and data aggregation over the WSN.

5.7. Energy-Aware Context Recognition Algorithm (EACRA)

The proposed Energy-Aware Context Recognition Algorithm for data acquisition minimizes the WSN node's energy consumption by using a minimum number of sensor modules with a low sampling rate [58]. As a result, this technique minimizes the data packet size, sampling rate, and the number of packets transmitted. In addition, it increases the sleep mode time of the sensor node. This solution dynamically configures sensors to send specific data under specific conditions and at a particular time, thus avoiding redundant data transmissions. This algorithm uses SEEB declared knowledge and forces the sensor node to send data only when the context changes.

6. Networking Layer

Energy conservation is one of the goals in the design of routing protocols for WSNs. In the case of static WSNs, the directly connected nodes to the gateway suffer due to the high network traffic. Additionally, when using the same path in forwarding data, the energy of the nodes constructing that path expires. As a result, we lose a set of nodes without balancing the load using another path to extend the network's lifetime. In this section, we investigate the energy-aware protocols at the data link layer and network layer level.

6.1. The Network Layer Protocols

One of the approaches to minimize data transmission is in-node and in-network data processing. It is more efficient to process data than transmit it, but data fusion complicates secure routing and introduces delay and unreliability. Another way is the use of hierarchical routing that helps to achieve energy efficiency in WSN. However, it may suffer from various issues such as optimum cluster size, cluster head selection, communication, cluster topology, and scalability. These issues become more challenging in the case of node failure where network topology changes, impacting network connectivity. Eventually, the majority of network energy will be consumed in cluster formation rather than data transmission. The multichannel routing approach allows sensors to use multiple orthogonal channels for sending data. Yes, it ensures parallel communications, decreases delay, and increases throughput in WSNs. However, it requires additional resources for channel scanning to decide upon available ones, channel decision to select a channel, and channel switching to switch from a channel to another one. For sure, this will require more energy to provide the needed Quality of Service (QoS) in WSN. In this section, we present the energy-aware routing protocols for WSN.

(1) Low-Energy Adaptive Clustering Hierarchy (LEACH)

It is a protocol with self-organizing and adaptive clustering. It randomly distributes the energy load among the sensors in the network. LEACH assumes that all WSN nodes have enough power to transmit data to reach the gateway. In this protocol, WSN nodes have enough computational power to support different MAC protocols. Nodes that are located close to each other have correlated data. A node called cluster-head (CH) in each cluster acts as the local base station. LEACH protocol randomly rotates the cluster head role between all WSN nodes in order to make activities equally shared among the WSN nodes; hence, they consume power equally. LEACH also performs data fusion when data is sent from the cluster heads to the base station, thus reducing energy consumption and extending network lifetime [50].

(2) Threshold-Sensitive Energy-Efficient Sensor Network Protocol (TEEN)

TEEN is a cluster-based hierarchical routing protocol for WSN that is based on LEACH protocol. TEEN is dedicated to time-critical applications, and it is based on two assumptions: first, the assumption that sensors and the base station have the same initial energy, and second, the assumption that the base station can transmit data to all nodes in the WSN without intermediate nodes. This protocol consists of simple nodes, first-level cluster heads, and second-level cluster heads. TEEN follows LEACH's strategy to form the cluster. Hence, first-level cluster heads are based far from the base station, and second-level cluster heads are based near the base station. The cluster head sends two types of data: the hard threshold and the soft threshold. The hard threshold represents data in the range of interest, reducing the number of packet transmissions. The soft threshold represents data out of the range of interest; hence, it is transmitted [59].

(3) Adaptive Periodic Threshold-Sensitive Energy-Efficient Sensor Network Protocol (APTEEN)

It is an improved version that has all the features of TEEN. It was developed for hybrid networks to capture periodic data collection and react to time-critical events [60]. APTEEN supports queries like historical analysis of past data values, snapshots of the current network view, and persistent monitoring of an event. After selecting the cluster head, the cluster head periodically broadcasts the following parameters: Attributes (interested physical parameters), Thresholds (hard threshold value and soft threshold value), Time schedule (time slot using TDMA), and the count time (the maximum period between two successive reports sent by a node). If a node does not send data during the count time, it must sense the environment and send the sampled data. Since it is a hybrid protocol, it can emulate a proactive network or a reactive network depending on the count time and threshold value. The disadvantage of this protocol is that additional complexity is needed to implement the threshold and count time.

(4) Power-Efficient Gathering in Sensor Information Systems (PEGASIS)

It is a near-optimal, chain-based, power-efficient protocol based on LEACH. In this protocol [61], each node has information about all other nodes. Additionally, each node can transmit data to the gateway directly. PEGASIS assumes that all WSN nodes have the same energy level and they will consume their power at the same time. Since all WSN nodes have information about the network status, the protocol can construct a chain using a greedy algorithm. The creation of the chain starts at a node that is away from the gateway. Each node sends and receives data from the closest node of its neighbors. The sensor node uses signal strength to measure the distance; it only adjusts the signal strength to reach the nearest node. When constructing the chain, each node merges received data with its data. In each round of chain construction, the protocol randomly selects a node to be a leader from the chain that will transmit the aggregated data to the gateway. PEGASIS outperforms LEACH as it eliminates the overhead of dynamic cluster information. In this protocol, each node requires global information about the network. Such a requirement is a drawback of this protocol because the network might change when a node goes down. As a result, nodes should be updated each time, introducing more overhead, thus more energy consumption.

(5) Sensor Protocols for Information via Negotiation (SPIN)

It is a set of adaptive protocols that uses data negotiation and resource-adaptive algorithms [62]. SPIN is data-centric, and it assumes that all WSN nodes in the network are gateways and that nodes close to each other have similar data. SPIN names sensor data using meta-data. Since all nodes are considered gateways, all information is sent to all nodes in the WSN. As a result, the user can query any node and get the data immediately. Nodes in this network use a high-level name to describe their collected data called meta-data. Therefore, before transmission, nodes exchange meta-data, hence avoiding the retransmission of redundant data sets. When receiving the data, the sensor node advertises it to its neighbors. Meta-data is used for negotiation to solve the flooding problem, hence improving energy efficiency of the WSN. SPIN uses three types of messages: ADV, REQ, and DATA for communication within the WSN. The ADV message is used for advertising new data, while the REQ message is used to request data. The DATA message transmits data sets. For example, when a node gets some new data and wants to distribute it throughout the network, it broadcasts an ADV message containing the meta-data. Only the interested nodes request that data by sending a REQ message. Then the node replies by a DATA message to the requesting sensor node.

The neighboring node keeps repeating this process until WSN nodes get the new data. SPIN has two main protocols, which are SPIN-1 and SPIN-2, which incorporate negotiations before transmitting data. Hence, only useful information will be sent. Each node in the WSN has its resource manager that keeps track of resource consumption. SPIN-1 incorporates negotiation, while SPIN-2 includes a threshold-based resource awareness mechanism. SPIN provides energy savings using meta-data negotiation, which reduces redundant data by 50%. However, the SPINs do not guarantee the delivery of data to the interested sensor nodes. For example, if nodes interested in the data are away from the source node and the intermediate nodes between the source and the destination are not interested in that data, such data will not be delivered to the destination.

(6) Rumor Routing (RR)

Rumor Routing combines query flooding and event flooding protocols. It has the following assumptions: The network is composed of densely distributed nodes where Links are bi-directional, and the distance of transmissions is short. This protocol floods the events with a small number but with a large number of queries. The query is forwarded to the particular node that belongs to the interested region. To flood events, the protocol uses packets (agents) that have a long lifetime. When a node detects an event, it adds it to its local table (events table) and generates an agent. The agent goes through the network with that event information. When a node needs to initiate a query, it routes the query to the initial source. If it gets some nodes on the path before its TTL expires, it will be routed to the event; else, the node will need to retransmit, give up, or flood the request. Rumor Routing only maintains one path between source and destination. It performs well if the number of events is small. In the case of many events, preserving agents and event tables in each node is costly if there is not enough interest in these events from the base station. In addition, the overhead associated with this routing protocol is controlled by different parameters such as TTL for agents and queries [63].

(7) Geographical Energy-Aware Routing (GEAR)

Routing location-based protocols need location information of all the WSN nodes in the network to calculate the distance between each pair of nodes. GEAR [64] is a location-based routing protocol that aims to locate nodes in the WSN. Each node stores two costs to reach the destination in this protocol: the estimated cost and learned cost. First, the estimated cost combines residual energy and the distance to the destination. Second, the learned cost is a modified estimated cost. When a node does not have any nearby neighbors towards the destination, we say there is a hole. When no holes exist, then the estimated cost equals the learned cost. The GEAR protocol only considers a certain region instead of sending the interests to the WSN; therefore, it restricts the number of interests. The protocol goes through two phases. In Phase-I, packets are forwarded towards the target region.

When receiving a packet, the sensor node searches for a neighbor close to the targeted region. If there are many suitable nodes, one node is picked as the next hop to forward the packet based on the learned cost function. In phase-II, the packets are forwarded within the region. When the packet reaches the region, it is diffused by recursive geographic forwarding or restricted flooding, depending on the density of nodes.

(8) Geographic Adaptive Fidelity (GAF)

GAF is an energy-efficient routing protocol that is location based [65]. It was conceived for mobile ad hoc networks, but it can also be applied to fixed sensor networks. However, GAF is a location-based protocol; it can also be implemented as a hierarchical protocol where clusters are based on geographic location. The area of interest is split into fixed zones where each node uses its GPS location to locate itself within the zone. Nodes of the same zone are considered equivalent in routing cost. Hence, GAF keeps nodes located in the same area in a sleeping state, which increases the WSN lifetime.

(9) Energy-Aware Routing Protocol (EARP)

It is a distributed energy-aware routing protocol. In this routing protocol, the network is divided into clusters [66]. Each cluster elects the head that will be directly connected to each node in its cluster. This election is based on node energy, where the selected node's energy ratio must be higher than the average energy ratio of the other nodes in the cluster. This way, the routing process will involve only the nodes that have high energy, as a result, extending the lifetime of the network.

Many energy-aware routing protocols have been proposed, such as AESC, ODYSSE, and GBR, [67–69]. However, in general, energy-efficient routing protocols are classified into the following four categories: first, intelligence-based categorization; second, structure-based categorization where energy-efficient data-centric, flat, hierarchical, and location-aware routing protocols are examined; third, the operation-based categorization discusses QoS and energy-efficient data delivery models; fourth, the partial energy-efficient categorization discusses some aspects of energy-efficient routing protocols in WSNs. These categories have been discussed in [70–85].

6.2. The Data Link Layer Protocols

Medium Access Control (MAC) protocols focused on throughput, with less emphasis on energy optimization. Recently, several MAC protocols have been designed to consider energy optimization. In WSN, the focus of most MAC protocols is to minimize the idle power consumption by setting the RF module into a sleep mode whenever possible. In this part, we discuss the different energy-aware MAC protocols.

(1) Sensor-MAC Protocol

S-MAC is among the first MAC protocols designed for WSN [86]. In SMAC, nodes create a schedule that decides when to activate their RF module and enter sleep mode. In this protocol, nodes must share four sleep schedules' information using periodic SYNC packets. When a node is willing to send a packet to another node, it waits until the destination wakes up and then sends the packet using CSMA/CA. S-MAC also supports a message-passing mechanism that breaks long packets into fragments. These fragments are sent and acknowledged (RTS-CTS exchange). Additionally, to avoid retransmission of long packets, fragments and ACKs can serve the purposes of the RTS and CTS packets, especially for nodes that wake up in the middle of transmission and have already missed the original RTS-CTS exchange.

(2) Timeout-MAC Protocol

T-MAC aims to eliminate idle energy by setting the length of the active portion of the frames [87]. Instead of sending messages during a predetermined active period, messages are forwarded in bursts mode at the beginning of the frame. After a certain amount of time, when no event occurs, sensor nodes put their RF modules into a sleep state and wait for the next scheduled active frame. The events include the firing of the frame timer, received data packets or transmitted data packets, the sensing of radio communication, neighboring sensors' data exchanges, or any radioactivity. When node A starts sending data to node B, considering that node B and node D did not detect node A's transmissions, they start sending data to A after a timeout. The data transmission from A to E is interrupted and continued during the coming active period. T-MAC is better because S-MAC requires a long active period to make sure that traffic is transmitted.

(3) D-MAC Protocol

In some data acquisition techniques, the time for packets to arrive at the gateway is mostly stable and can be predictable [88]. DMAC takes this advantage and wakes up nodes based on their distance from the gateway. The wake-up scheme is composed of a receiving period and a sending period that are needed for a single transmission (period = P). This period is followed by a long sleep time. The nodes on the data acquisition tree begin the receiving period after T, where T = D $*$ P, and D is the depth of the node on the tree. Hence, a node's receiving period goes with its neighbor's send period.

(4) Traffic-Adaptive Medium Access (TRAMA)

TRAMA tries to reduce the energy that is wasted by packet collisions [89]. In the beginning, nodes exchange information with each other via a Neighbor Protocol (NP). Afterward, nodes transmit the schedule and data packets via the Schedule Exchange Protocol (SEP). Based on the neighbor traffic and schedule information, nodes determine their priorities. When two nodes are willing to communicate, the node with the highest priority sends its data first.

(5) Sparse Topology and Energy Management (STEM)

In STEM, all sensor nodes go into sleep mode while monitoring the environment [90]. They wake up only when traffic is generated and ready to be sent. The STEM-T version uses a separate channel to awaken neighboring nodes, while in the STEM-B version, the node sends beacons on the control channel so sleeping nodes switch on their RF modules to receive the messages. The STEM-T guarantees a minimum delay because receivers are turned on immediately after data are ready, but it requires more overhead regarding the STEM-B. STEM-T might also require extra hardware such as the RF module.

6.3. Network Topologies for WSN

Implementing a WSN can be in different topologies such as Star, Tree, Mesh, Cluster, and Grid topology. Here we discuss the impact of topologies and routing protocols used in WSN on energy consumption and the network lifetime (Tables 6 and 7).

Table 6. The usage of data acquisition techniques in different topologies.

Topology Data Acquisition Technique	Star	Tree	Mesh	Cluster
Directed diffusion	YES	YES	YES	YES
Chain construction	NO	YES	YES	YES (Between cluster heads)
Probabilistic model	YES	YES	YES	YES
Data correlation	NO	NO	NO	YES
Query model	NO	YES	YES	YES

Table 7. The usage of energy-aware routing protocols in different topologies.

Topology / Routing Protocol	Star	Tree	Mesh	Cluster
LEACH	NO	NO	NO	YES
TEEN	NO	NO	NO	YES
APTEEN	NO	NO	NO	YES
PEGASIS	NO	YES	YES	NO
SPIN	NO	NO	YES	NO
RR	NO	NO	YES	YES
GEAR	NO	NO	YES	YES
GAF	NO	NO	YES	YES
EAP	NO	NO	NO	YES
GBR	NO	NO	NO	YES
AESC	NO	YES	YES	YES
PERP	NO	YES	YES	YES
ODYSSE	NO	YES	YES	YES

(1) Star Topology

This is the easiest network topology to implement. In this topology, nodes are connected to the base station and cannot communicate directly. It has the disadvantage of single path communication; hence, there is no backup if the link to the base station is lost. Additionally, if a node wants to reach the base station far from it, it needs to generate a high-power signal. In such a topology, we cannot benefit from several data acquisition techniques such as data aggregation.

(2) Tree Topology

In this topology, WSN nodes form a tree where the gateway is the root of that tree. The path from any node to the gateway may be single-hop or multi-hop. Such topology allows for the usage of data acquisition techniques and energy-aware routing protocols. However, as there is always a single path from any node to the gateway, all the sub-nodes get disconnected from the network if one node in the path fails.

(3) Mesh Topology

The Mesh topology presents several advantages for WSN. On the one hand, the network is a full mesh where each node is connected to all other nodes. On the other hand, when each node is only connected to some other nodes, the network is called partial mesh. This topology has the advantages of self-healing, self-optimization, self-configuration, and self-protection, which are critical requirements in the design of WSN.

(4) Cluster Topology

In this topology, the WSN network is divided into clusters. At any moment, there should be at least one node working in each cluster, and this node is called the cluster head. The head is responsible for transmitting data packets and routing information from the subsequent nodes to the gateway.

The topology choice directly affects the data acquisition technique that can be used.

The authors in [91–93] concluded that the energy consumption depends on the WSN network topology. This energy consumption is due to network joining, route discovery, route update, and network self-healing. Therefore, the network designer should consider the number of sensor nodes deployed before considering a specific network topology. If the number of sensor nodes is low, the star topology will serve as the optimal topology since it offers better performance and reliability with low energy consumption. However, as the number of nodes increases, the designer should consider cluster tree topology as it divides the whole WSN network into zones where each cluster head manages its zone.

7. Energy Storage in IoT Devices

In WSN nodes, the source of energy is either batteries, energy harvesting, or both. While designing WSN nodes, batteries must be small in size, and, thus, the available energy resources will be limited. At the same time, WSN nodes need to stay alive as long as possible.

There are different battery types available for WSN nodes in the market. However, several criteria must be considered when comparing them, such as cycle life, environment, safety, output voltage, and energy density. In this section, we present the different batteries used to power IoT devices in SEEB [94].

7.1. Nickel-Cadmium Battery

Nickel-cadmium batteries are commonly used in buildings to power various devices. Each nickel-cadmium cell contains a cadmium anode, nickel-hydroxide cathode, and alkaline electrolyte. They offer high currents with constant voltage, are tolerant of physical abuse, and have inefficient usage cycling. Nickel-cadmium cells are expensive and toxic, which makes this type of battery not environmentally friendly.

7.2. Nickel Metal Hydride (NiMH)

Like a rechargeable battery, the main difference compared to the previous one is that it maintains its characteristics through many recharges (life cycle). It is heavy with lower energy density, so the NiMH AA battery begins with 1.2 V instead of the conventional 1.5 V as for an alkaline AA battery. It does not drain fast as the current draw increases. Additionally, it has a wide working temperature range, making it deployable in low- and high-temperature environments where lithium-based batteries are prone to explode.

7.3. Alkaline

Alkaline batteries are the most commonly used in powering WSN nodes. Since they have a fixed energy rating, their lifetime is limited. For example, a node operating at a 1% duty cycle on standard 3000 mAh AA batteries would require battery replacement every 82 weeks. They have higher energy density compared to zinc-carbon batteries, and the output voltage of each cell is about 1.5 V. Therefore, to get higher voltage, cells should be arranged in series. Reusable alkaline batteries are not efficient compared to nickel-cadmium ones, and their energy density decreases dramatically with cycling.

7.4. Zinc-Carbon

Due to their low cost, Zinc-carbon batteries are widely used where each cell outputs 1.5 V. Besides being not rechargeable, their main drawback is the outer shell made of zinc, which serves as the anode for the cell. Hence, it creates holes that leak the acidic electrolyte, which might damage the powered device. As a result, the usage of this type of battery should not be used in critical applications.

7.5. Lithium Polymer

The cells of this battery type are five times greater than the lead-acid ones and three times greater than the cells of alkaline batteries. In terms of size, lithium polymer batteries are lighter and thinner. They start producing at a voltage of 3.0 V with a stable voltage supply. However, they are dangerous compared to lead-acid cells and alkaline batteries since they explode when lithium contacts water. Therefore, they are small in size. Rechargeable lithium cells provide 3.7 V. In WSN applications, alkaline batteries are preferred due to their low cost, high capacity, and low self-discharge rate.

7.6. Energy Autonomy

Nowadays, energy autonomy is the ideal technique to power IoT devices. Energy autonomy can be achieved through different mechanisms such as hybrid-harvesting technology, energy harvesting (EH) techniques, micro-fabrication technologies, and reinforcement learning techniques. In [95,96], the authors discussed different energy autonomy approaches used in WSN, such as thermoelectric antennas, solar antennas, electromagnetic energy harvesters, and multi-tone signals. The researchers concluded that energy harvesting with renewable energy sources is predicted to be the future power of IoT devices.

8. Proposed Framework to Design Energy-Aware WSN for SEEB

This paper conveyed that energy optimization in WSN nodes can be achieved at different levels, such as the hardware design, the operating system choice, the data acquisition technique, and the networking technology. The next figure illustrates the proposed framework for designing a WSN inside a smart building.

Figure 3 illustrates how the deployment environment affects the choice of the WSN node's components. First, the deployment environment directly affects the network topology because of the interference and connectivity issues indoors. For example, opting for a full mesh network topology, where obstacles might not allow a node to be connected with multiple nodes, will not be a good solution. As a result, the topology choice will impact the node's role in the network (end node, router node, or cluster-head). Second, the wireless sensor network goes for a specific routing protocol and a medium access control protocol depending on the network topology. The network topology also affects the data acquisition technique at the application layer. Hence, the adopted operating system must support the selected network protocol, data link protocol, and the data acquisition technique.

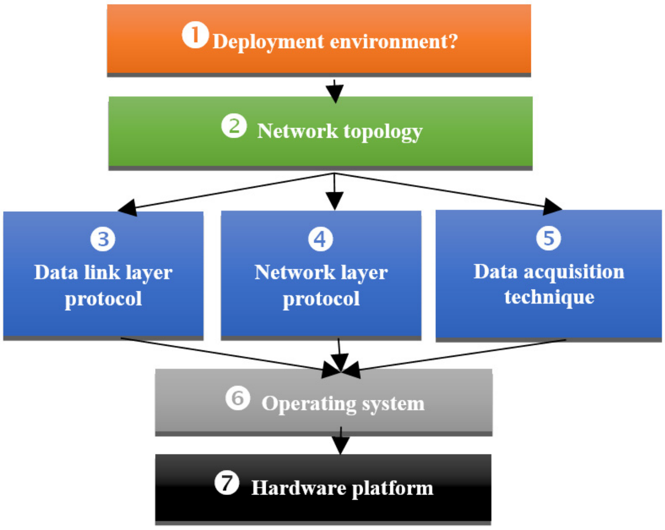
❶ Deployment environment?
❷ Network topology
❸ Data link layer protocol
❹ Network layer protocol
❺ Data acquisition technique
❻ Operating system
❼ Hardware platform

Figure 3. WSN node components driven by deployment environment.

Moreover, the selected hardware configuration must support the operating system. The battery choice is also based on power capacity and the deployment environment, as each battery type is dedicated to a specific deployment environment. For example, lithium polymer cannot be used in an environment that might contact water or high humidity.

8.1. Deployment Environment

The deployment environment has a great impact on the WSN as a whole. Depending on the building's architecture, there might be different scenarios of the number of nodes, radio frequency interference, network connectivity, and accessibility. In addition, the building's infrastructure might impact the energy source of the sensor node. The building's architecture and the number of nodes needed will bring up one main issue: node connectivity. Such an issue will mandate a specific network topology to overcome the pertaining networking challenges.

8.2. Network Topology

The choice of the network topology is mainly affected by the deployment environment. The selected network topology is judged as the ideal network topology for the WSN based on data acquisition technique, the data link layer, and the network layer routing protocol, ensuring successful packet transmission with low power consumption.

8.3. Data Link Layer Protocol

The network topology affects the choice of the data link layer protocol. A robust network topology requires the selection of the data link protocols, which can go with the RF modules and provide the best coordination with the network layer protocol, along with low power consumption.

8.4. Network Layer Protocol

The network topology also affects the choice of the network layer protocol. A robust network topology requires the selection of the network protocols, which can provide the best network self-healing and node management along with low power consumption.

8.5. Data Acquisition Technique

The data acquisition technique is mainly affected by the choice of the network topology. For instance, a data acquisition technique that provides data aggregation might require a specific network topology to gather data in a centralized point before forwarding it to the gateway of the WSN.

8.6. Operating System

As data acquisition techniques, network layer protocols, and data layers protocols can be supported by a specific operating system, the operating system choice must provide and fulfill the above requirements.

8.7. Hardware Platform

The selected operating system needs a specific hardware configuration. The hardware platform must support different features of the operating system, mainly the operating system's architecture, and provide good performance without system failure or delays in response time.

9. Conclusions

In this paper, we presented the state of the art of low-power technologies used in designing a WSN for indoor applications, e.g., SEEB. We surveyed the different relevant hardware components, operating systems, networking protocols, data acquisition techniques, and battery types in this context. As a result, we compared energy-aware data acquisition techniques and operating systems that can host these techniques. In addition, we presented the different networking protocols that are energy aware along with their network topologies and operating systems. Additionally, we compared different hardware components such as microcontrollers and antennas in terms of energy consumption. The paper also presented different battery types and energy sources that can power IoT devices. Hence, the paper serves as a reference for researchers when investigating the various energy optimization opportunities in IoT devices for SEEB.

Moreover, we proposed a framework for designing WSN in smart buildings along with the appropriate WSN configuration for optimal energy consumption. Additionally, we investigated convenient WSN node reliability and connectivity. The proposed framework can serve as a blueprint for researchers to deploy better adaptive solutions that combine energy-aware technologies, approaches, and hardware platforms to develop convenient WSN nodes for SEEB. This stated, our proposed framework responds to needed network connectivity and coverage challenges in indoor environments.

Author Contributions: Conceptualization, N.N., M.R.A., N.K., and D.B.; methodology, N.N., M.R.A., N.K., and D.B.; software, N.N.; validation, M.R.A., N.K., and D.B.; formal analysis, N.N., M.R.A., and D.B.; investigation, N.N., M.R.A., and D.B.; resources, N.N., M.R.A., and D.B.; data curation, N.N.; writing—original draft preparation, N.N.; writing—review and editing, M.R.A., and D.B.; visualization, N.N., M.R.A., and D.B.; supervision, M.R.A., N.K., and D.B.; project administration, M.R.A., and D.B.; funding acquisition, M.R.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the US-NAS/USAID under the PEER Cycle5 project grant number #5-398 entitled "Towards Smart MG: Renewable Energy Integration into Smart Buildings".

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kanoun, O.; Bradai, S.; Khriji, S.; Bouattour, G.; El Houssaini, D.; Ben Ammar, M.; Naifar, S.; Bouhamed, A.; Derbel, F.; Viehweger, C. Energy-Aware System Design for Autonomous Wireless Sensor Nodes: A Comprehensive Review. *Sensors* **2021**, *21*, 548. [CrossRef] [PubMed]
2. Priyadarshi, R.; Gupta, B.; Anurag, A. Deployment techniques in wireless sensor networks: A survey, classification, challenges, and future research issues. *J. Supercomput.* **2020**, *76*, 7333–7373. [CrossRef]
3. Kandris, D.; Nakas, C.; Vomvas, D.; Koulouras, G. Applications of Wireless Sensor Networks: An Up-to-Date Survey. *Appl. Syst. Innov.* **2020**, *3*, 14. [CrossRef]
4. Singh, P.K.; Bhargava, B.K.; Paprzycki, M.; Kaushal, N.C.; Hong, W.C. (Eds.) *Handbook of Wireless Sensor Networks: Issues and Challenges in Current Scenario's*; Springer: Berlin/Heidelberg, Germany, 2020.
5. Boaventura, A.S.; Carvalho, N.B. A low-power wake-up radio for application in WSN-based indoor location systems. *Int. J. Wirel. Inform. Netw.* **2013**, *20*, 67–73. [CrossRef]
6. Gubbi, J.; Buyya, R.; Marusic, S.; Palaniswami, M. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Gener. Comput. Syst.* **2013**, *29*, 1645–1660. [CrossRef]
7. Gravina, R.; Alinia, P.; Ghasemzadeh, H.; Fortino, G. Multi-sensor fusion in body sensor networks: State-of-the-art and research challenges. *Inf. Fusion* **2017**, *35*, 68–80. [CrossRef]
8. Berthier, F.; Beigne, E.; Vivet, P.; Sentieys, O. Power gain estimation of an event-driven wake-up controller dedicated to WSN's microcontroller. In Proceedings of the IEEE 13th International New Circuits and Systems Conference (NEWCAS), Grenoble, France, 7–10 June 2015; pp. 1–4.
9. Karray, F.; Jmal, M.W.; Garcia-Ortiz, A.; Abid, M.; Obeid, A.M. A comprehensive survey on wireless sensor node hardware platforms. *Comput. Netw.* **2018**, *144*, 89–110. [CrossRef]
10. Selmic, R.R.; Phoha, V.V.; Serwadda, A. *Wireless Sensor Networks*; Springer International Publishing AG: New York, NY, USA, 2016.
11. Rashid, B.; Rehmani, M.H. Applications of wireless sensor networks for urban areas: A survey. *J. Netw. Comput. Appl.* **2016**, *60*, 192–219. [CrossRef]
12. Engmann, F.; Katsriku, F.A.; Abdulai, J.-D.; Adu-Manu, K.S.; Banaseka, F.K. Prolonging the Lifetime of Wireless Sensor Networks: A Review of Current Techniques. *Wirel. Commun. Mob. Comput.* **2018**, *2018*, 1–23. [CrossRef]
13. Kumar, N.; Singh, Y. Routing protocols in wireless sensor networks. In *Handbook of Research on Advanced Wireless Sensor Network Applications, Protocols, and Architectures*; IGI Global: Hershey, PA, USA, 2017; pp. 86–128.
14. Li, W.; Kara, S. Methodology for Monitoring Manufacturing Environment by Using Wireless Sensor Networks (WSN) and the Internet of Things (IoT). *Procedia CIRP* **2017**, *61*, 323–328. [CrossRef]
15. Mysore, S.; Agrawal, B.; Chong, F.T.; Sherwood, T. Exploring the Processor and ISA Design for Wireless Sensor Network Applications. In Proceedings of the 21st International Conference on VLSI Design (VLSID 2008), Hyderabad, India, 4–8 January 2008; pp. 59–64. [CrossRef]
16. Abu Alsheikh, M.; Lin, S.; Niyato, D.; Tan, H.-P. Rate-Distortion Balanced Data Compression for Wireless Sensor Networks. *IEEE Sens. J.* **2016**, *16*, 5072–5083. [CrossRef]
17. Yussoff, Y.; Abidin, H.Z.; Rahman, R.A.; Yahaya, F.H. Development of a PIC-based wireless sensor node utilizing XBee technology. In Proceedings of the 2nd IEEE International Conference on Information Management and Engineering, Chengdu, China, 16–18 April 2010; pp. 116–120. [CrossRef]
18. Song, Y.; Gong, C.; Feng, Y.; Ma, J.; Zhang, X. Design of Greenhouse Control System Based on Wireless Sensor Networks and AVR Microcontroller. *J. Netw.* **2011**, *6*, 1668. [CrossRef]
19. Handziski, V.; Polastre, J.; Hauer, J.H.; Sharp, C. Flexible hardware abstraction of the TI MSP430 mi-crocontroller in TinyOS. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*; ACM: New York, NY, USA, 2004; pp. 277–278.
20. Costa, F.M.; Ochiai, H. Energy-Efficient Physical Layer Design for Wireless Sensor Network Links. In Proceedings of the 2011 IEEE International Conference on Communications (ICC), Kyoto, Japan, 5–9 June 2011; pp. 1–5. [CrossRef]

21. Han, D.-M.; Lim, J.-H. Smart home energy management system using IEEE 802.15.4 and zigbee. *IEEE Trans. Consum. Electron.* **2010**, *56*, 1403–1410. [CrossRef]
22. Boonsawat, V.; Ekchamanonta, J.; Bumrungkhet, K.; Kittipiyakul, S. XBee wireless sensor networks for temperature monitoring. In Proceedings of the 2nd Conference on Application Research and Development (ECTI-CARD 2010), Chon Buri, Thailand, 5–8 May 2016.
23. XBee. XBee-PRO™ OEM RF Modules. In *Product Manual v1. xAx-802.15, 4*; XBee: Hopkins, MN, USA, 2013.
24. Schurgers, C.; Aberthorne, O.; Srivastava, M. Modulation scaling for energy aware communication systems. In *Proceedings of the 2001 International Symposium on Low Power Electronics and Design*; ACM: New York, NY, USA, 2001; pp. 96–99.
25. Yang, K.; Song, L. AMC for TwoWayAmplify-and-Forward Relay Networks. In Proceedings of the IEEE International Conference on Communications, Xiamen, China, 28–30 July 2021.
26. Kim, D.; Jung, B.C.; Lee, H.; Sung, D.K.; Yoon, H. Optimal Modulation and Coding Scheme Selection in Cellular Networks with Hybrid-ARQ Error Control. *IEEE Trans. Wirel. Commun.* **2008**, *7*, 5195. [CrossRef]
27. Chen, S.; Zhao, C.; Wu, M.; Sun, Z.; Zhang, H.; Leung, V.C. Compressive network coding for wireless sensor net-works: Spatio-temporal coding and optimization design. *Comput. Netw.* **2016**, *108*, 345–356. [CrossRef]
28. Mazloum, N.S.; Edfors, O. Comparing analog front-ends for duty-cycled wake-up receivers in wireless sensor net-works. *IEEE Sens. J.* **2016**, *16*, 7016–7021. [CrossRef]
29. Middya, R.; Chakravarty, N.; Naskar, M.K. Compressive sensing in wireless sensor networks—A survey. *IEEE Tech. Rev.* **2017**, *34*, 642–654. [CrossRef]
30. Mukesh, S.; Iqbal, M.; Jianhua, Z.; Ping, Z.; Rehman, I.U. Comparative Analysis of M-ary Modulation Techniques for Wireless Ad-hoc Networks. In Proceedings of the 2007 IEEE Sensors Applications Symposium, San Diego, CA, USA, 6–8 February 2007; pp. 1–6. [CrossRef]
31. Howard, S.L.; Schlegel, C.; Iniewski, K. Error Control Coding in Low-Power Wireless Sensor Networks: When Is ECC Energy Efficient? *EURASIP J. Wireless Commun. Netw.* **2006**, *2006*, 74812. [CrossRef]
32. Kadel, R.; Paudel, K.; Guruge, D.B.; Halder, S.J. Opportunities and Challenges for Error Control Schemes for Wireless Sensor Networks: A Review. *Electronics* **2020**, *9*, 504. [CrossRef]
33. Yang, J.; Khandani, A.K.; Tin, N. *Adaptive Modulation and Coding in 3G Wireless Systems*; Technical Report Number UWE&CE#2002-15; IEEE: Piscataway, NJ, USA, 2002.
34. Morelos-Zaragoza, R.H.; Fossorier, M.P.C.; Imai, H. Multilevel Coded Modulation for Unequal Error Protection and Multistage Decoding—Part I: Symmetric Constellations. *IEEE Trans. Commun.* **2000**, *48*, 2. [CrossRef]
35. Salih, S.H.O.; Suliman, M.M.A. Implementation of AMC Technique. *Int. J. Sci. Eng. Res.* **2011**, *2*, 5.
36. Islam, M.R. Error Correction Codes in Wireless Sensor Network: An Energy Aware approach. *Int. J. Comput. Inform. Eng.* **2010**, *4*, 1.
37. Yousof Naderi, M.; Rabiee, H.R.; Khansari, M. Performance Analysis of Selected Error Control Protocols in Wireless Multimedia Sensor Networks. In Proceedings of the 18th Annual IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, Miami Beach, FL, USA, 17–19 August 2010.
38. Wang, Z.; Chiang, P.-Y.; Nazari, P.; Wang, C.-C.; Chen, Z.; Heydari, P. A CMOS 210-GHz Fundamental Transceiver with OOK Modulation. *IEEE J. Solid-State Circuits* **2014**, *49*, 564–580. [CrossRef]
39. Hahm, O.; Baccelli, E.; Petersen, H.; Tsiftes, N. Operating Systems for Low-End Devices in the Internet of Things: A Survey. *IEEE Internet Things J.* **2016**, *3*, 720–734. [CrossRef]
40. Sharma, D. Low Cost Experimental set up for Real Time Temperature, Humidity Monitoring through WSN. *Int. J. Eng. Sci.* **2017**, *7*, 4340.
41. Amjad, M.; Sharif, M.; Afzal, M.K.; Kim, S.W. TinyOS-new trends, comparative views, and supported sensing applications: A review. *IEEE Sens. J.* **2016**, *16*, 2865–2889. [CrossRef]
42. Durmaz, C.; Challenger, M.; Dagdeviren, O.; Kardas, G. *Modelling Contiki-Based IoT Systems*; OA-SIcs-OpenAccess Series in Informatics; Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik: Wadern, Germany, 2017; Volume 56.
43. Bhatti, S.; Carlson, J.; Dai, H.; Deng, J.; Rose, J.; Sheth, A.; Shucker, B.; Gruenwald, C.; Torgerson, A.; Han, R. MANTIS OS: An Embedded Multithreaded Operating System for Wireless Micro Sensor Platforms. *Mob. Netw. Appl.* **2005**, *10*, 563–579. [CrossRef]
44. Eswaran, A.; Rowe, A.; Rajkumar, R. Nano-rk: An energy-aware resource-centric rtos for sensor net-works. In Proceedings of the 26th IEEE International Real-Time Systems Symposium (RTSS'05), Miami, FL, USA, 5–8 December 2005; p. 10.
45. Cao, Q.; Abdelzaher, T. LiteOS: A lightweight operating system for C++ software development in sensor networks. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*; ACM: New York, NY, USA, 2006; pp. 361–362.
46. Intanagonwiwat, C.; Govindan, R.; Estrin, D. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*; ACM: New York, NY, USA, 2000; pp. 56–67.
47. Du, K.; Wu, J.; Zhou, D. *Chain-Based Protocols for Data Broadcasting and Gathering in the Sensor Networks*; IEEE: Piscataway, NJ, USA, 2003; p. 8. [CrossRef]
48. Chen, J.; Li, J.; He, S.; Sun, Y.; Chen, H.-H. Energy-Efficient Coverage Based on Probabilistic Sensing Model in Wireless Sensor Networks. *IEEE Commun. Lett.* **2010**, *14*, 833–835. [CrossRef]

49. Jiang, H.; Jin, S.; Wang, C. Prediction or Not? An Energy-Efficient Framework for Clustering-Based Data Collection in Wireless Sensor Networks. *IEEE Trans. Parallel Distrib. Syst.* **2010**, *22*, 1064–1071. [CrossRef]
50. Chipara, O.; Lu, C.; Roman, G.-C. Real-Time Query Scheduling for Wireless Sensor Networks. In *28th IEEE International Real-Time Systems Symposium (RTSS 2007)*; IEEE: Piscataway, NJ, USA, 2007; pp. 389–399. [CrossRef]
51. Singh, K. WSN LEACH based protocols: A structural analysis. In *2015 International Conference and Workshop on Computing and Communication (IEMCON)*; IEEE: Piscataway, NJ, USA, 2015; pp. 1–7. [CrossRef]
52. Madden, S.; Franklin, M.J.; Hellerstein, J.M.; Hong, W. TAG: A tiny aggregation service for ad-hoc sensor networks. *ACM SIGOPS Oper. Syst. Rev.* **2002**, *36*, 131–146. [CrossRef]
53. Mahajan, S.; Dhiman, P.K. Clustering in Wireless Sensor Networks: A Review. *Int. J. Adv. Res. Comput. Sci.* **2016**, *7*, 3.
54. Ambigavathi, M.; Sridharan, D. Energy-Aware Data Aggregation Techniques in Wireless Sensor Network. In *Advances in Power Systems and Energy Management*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 165–173. [CrossRef]
55. Incel, O.D.; Krishnamachari, B. Enhancing the Data Collection Rate of Tree-Based Aggregation in Wireless Sensor Networks. In Proceedings of the 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, San Francisco, CA, USA, 16–20 June 2008; pp. 569–577. [CrossRef]
56. Qing, L.; Zhu, Q.; Wang, M. Design of a distributed energy-efficient clustering algorithm for heterogeneous wireless sensor networks. *Comput. Commun.* **2006**, *29*, 2230–2237. [CrossRef]
57. Vikash; Mishra, L.; Varma, S. Middleware Technologies for Smart Wireless Sensor Networks towards Internet of Things: A Comparative Review. *Wirel. Pers. Commun.* **2021**, *116*, 1539–1574. [CrossRef]
58. Naji, N.; Abid, M.R.; Krami, N.; Benhaddou, D. An Energy-Aware Wireless Sensor Network for Data Acquisition in Smart Energy Efficient Building. In *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*; IEEE: Piscataway, NJ, USA, 2019; pp. 7–12.
59. Manjeshwar, A.; Agrawal, D. TEEN: A routing protocol for enhanced efficiency in wireless sensor networks. In Proceedings of the15th International Parallel and Distributed Processing Symposium, IPDPS 2001, San Francisco, CA, USA, 23–27 April 2001; pp. 2009–2015. [CrossRef]
60. Manjeshwar, A.; Agrawal, D.P. APTEEN: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks. In *Proceedings of the International Parallel and Distributed Processing Symposium (IPDPSí)*; IEEE: Piscataway, NJ, USA, 2002; p. 195.
61. Lindsey, S.; Raghavendra, C. PEGASIS: Power-efficient gathering in sensor information systems. In *IEEE Aerospace Conference*; IEEE: Piscataway, NJ, USA, 2003. [CrossRef]
62. Jing, L.; Liu, F.; Li, Y. Energy saving routing algorithm based on SPIN protocol in WSN. In *2011 International Conference on Image Analysis and Signal Processing*; IEEE: Piscataway, NJ, USA, 2011; pp. 416–419. [CrossRef]
63. Braginsky, D.; Estrin, D. Rumor routing algorthim for sensor networks. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*; ACM: New York, NY, USA, 2002; pp. 22–31.
64. Nadeem, Q.; Rasheed, M.B.; Javaid, N.; Khan, Z.A.; Maqsood, Y.; Din, A. M-GEAR: Gateway-based energy-aware multi-hop routing protocol for WSNs. In *2013 Eighth International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA)*; IEEE: Piscataway, NJ, USA, 2013; pp. 164–169.
65. Roychowdhury, S.; Patra, C. Geographic Adaptive Fidelity and Geographic Energy Aware Routing in Ad Hoc Routing. *Int. J. Comput. Commun. Technol.* **2011**, *2*, 91–95. [CrossRef]
66. Liu, M.; Cao, J.; Chen, G.; Wang, X. An Energy-Aware Routing Protocol in Wireless Sensor Networks. *Sensors* **2009**, *9*, 445–462. [CrossRef]
67. Le, H.P.; John, M.; Pister, K. *Energy-Aware Routing in Wireless Sensor Networks with Adaptive Energy-Slope Control*; EE290Q-2 Spring; IEEE: Piscataway, NJ, USA, 2009.
68. Amdouni, I.; Adjih, C.; Aitsaadi, N.; Muhlethaler, N. *ODYSSE: A Routing Protocol for Wireless Sensor Networks. RR-8873, UPEC*; Inria Saclay; Inria: Paris, France, 2016.
69. Yoo, H.; Shim, M.; Kim, D.; Kim, K.H. GLOBAL: A Gradient-based routing protocol for load-balancing in large-scale wireless sensor networks with multiple sinks. In Proceedings of the IEEE Symposium on Computers and Communications, Riccione, Italy, 22–25 June 2010; pp. 556–562. [CrossRef]
70. Shirazi, S.S.; Haqiqat, A.T. Energy Efficient Hierarchical Cluster-Based Routing for Wireless Sensor Networks. *Int. J. Comput. Sci. Netw. Secur.* **2016**, *16*, 115.
71. Faheem, M.; Gungor, V. Energy efficient and QoS-aware routing protocol for wireless sensor network-based smart grid applications in the context of industry 4. *Appl. Soft Comput.* **2018**, *68*, 910–922. [CrossRef]
72. Khoufi, I.; Minet, P.; Laouiti, A.; Mahfoudh, S. Survey of deployment algorithms in wireless sensor networks: Coverage and connectivity issues and challenges. *Int. J. Autonomous Adapt. Commun. Syst.* **2017**, *10*, 341–390. [CrossRef]
73. Maimour, M.; Zeghilet, H.; Lepage, F. *Sustainable Wireless Sensor Networks*; Cluster-based Routing Protocols for Energy-Efficiency in Wireless Sensor Networks; Intech: Vienna, Austria, 2010; pp. 167–188.
74. Wei, C.; Yang, J.; Gao, Y.; Zhang, Z. Cluster-based Routing Protocols in Wireless Sensor Networks: A Survey. In Proceedings of the International Conference on Computer Science and Network Technology (ICCSNT '11), Harbin, China, 24–26 December 2011; pp. 1659–1663.
75. Guleria, K.; Verma, A.K. Comprehensive review for energy efficient hierarchical routing protocols on wireless sensor networks. *Wirel. Netw.* **2019**, *25*, 1159–1183. [CrossRef]

76. Anasane, A.A.; Satao, R.A. A Survey on Various Multipath Routing Protocols in Wireless Sensor Networks. *Procedia Comput. Sci.* **2016**, *79*, 610–615. [CrossRef]
77. Liu, X. A Survey on Clustering Routing Protocols in Wireless Sensor Networks. *Sensors* **2012**, *12*, 11113–11153. [CrossRef]
78. Manap, Z.; Ali, B.M.; Ng, C.K.; Noordin, N.K.; Sali, A. A Review on Hierarchical Routing Protocols for Wireless Sensor Networks. *Wirel. Pers. Commun.* **2013**, *72*, 1077–1104. [CrossRef]
79. Gherbi, C.; Aliouat, Z.; Benmohammed, M. A survey on clustering routing protocols in wireless sensor networks. *Sens. Rev.* **2017**, *37*, 12–25. [CrossRef]
80. Sikander, G.; Zafar, M.H.; Raza, A.; Babar, M.I.; Mahmud, S.A.; Khan, G.M. A Survey of Cluster-based Routing Schemes for Wireless Sensor Networks. *Smart Comput. Rev.* **2013**, *3*, 261–275. [CrossRef]
81. Singh, S.P.; Sharma, S. A Survey on Cluster Based Routing Protocols in Wireless Sensor Networks. *Procedia Comput. Sci.* **2015**, *45*, 687–695. [CrossRef]
82. Mehmood, A.; Song, H. Smart Energy Efficient Hierarchical Data Gathering Protocols for Wireless Sensor Networks. *Smart Comput. Rev.* **2015**, 425–462. [CrossRef]
83. Liu, X. Atypical Hierarchical Routing Protocols for Wireless Sensor Networks: A Review. *IEEE Sens. J.* **2015**, *15*, 5372–5383. [CrossRef]
84. Abid, R.M.; Saad, B. An interference-aware routing metric for Wireless Mesh Networks. *Int. J. Mob. Commun.* **2011**, *9*, 619. [CrossRef]
85. Abid, M.R. AAI3446193—Link Quality Characterization in IEEE 802.11s Wireless Mesh Networks. Ph.D. Thesis, Auburn University, Auburn, AL, USA, 2010.
86. Ye, W.; Heidemann, J.; Estrin, D. An energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the INFOCOM 21st Annual Joint Conference of the IEEE Computer and Communications Societies*; IEEE: Piscataway, NJ, USA, 2002; Volume 3, pp. 1567–1576.
87. Van Dam, T.; Langendoen, K. An adaptive energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*; ACM: New York, NY, USA, 2003; pp. 171–180.
88. Lu, G.; Krishnamachari, B.; Raghavendra, C. An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks. In *18th International Parallel and Distributed Processing Symposium*; IEEE: Piscataway, NJ, USA, 2004; p. 224. [CrossRef]
89. Ye, W.; Heidemann, J.; Estrin, D. Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks. *IEEE/ACM Trans. Netw.* **2004**, *12*, 493–506. [CrossRef]
90. Schurgers, C.; Tsiatsis, V.; Srivastava, M. STEM: Topology management for energy efficient sensor networks. In *IEEE Aerospace Conference*; IEEE: Piscataway, NJ, USA, 2003; Volume 3. [CrossRef]
91. Linden, D. *Handbook of Batteries and Fuel Cells*; McGraw-Hill Book Co.: New York, NY, USA, 1984; 1075p.
92. Varghese, S.G.; Kurian, C.P.; George, V.I.; John, A.; Nayak, V.; Upadhyay, A. Comparative study of zigBee topologies for IoT-based lighting automation. *IET Wirel. Sens. Syst.* **2019**, *9*, 201–207. [CrossRef]
93. Ari, D.; Cibuk, M.; Aggun, F. The Comparison of Energy Consumption of Different Topologies in Multi-hop Wireless Sensor Networks. In *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*; IEEE: Piscataway, NJ, USA, 2018; pp. 1–5. [CrossRef]
94. Buk, M.; Cengiz, M.S. Determination of Energy Consumption According to Wireless Network Topologies in Grid-Free Lighting Systems. *Light Eng.* **2020**, *28*, 2.
95. Gawali, S.K.; Deshmukh, M.K. Energy Autonomy in IoT Technologies. *Energy Procedia* **2019**, *156*, 222–226. [CrossRef]
96. Ashraf, S.; Saleem, S.; Ahmed, T.; Aslam, Z.; Muhammad, D. Conversion of adverse data corpus to shrewd output using sampling metrics. *Vis. Comput. Ind. Biomed. Art* **2020**, *3*, 1–13. [CrossRef] [PubMed]