

## Research Article

# Energy-Efficient Sleep/Wake Scheduling for Acoustic Localization Wireless Sensor Network Node

ChengFang Zhen,<sup>1</sup> Wenyi Liu,<sup>2</sup> Yongrui Liu,<sup>1</sup> and Anbin Yan<sup>1</sup>

<sup>1</sup> Key Laboratory of Instrumentation Science & Dynamic Measurement, North University of China, Ministry of Education, TaiYuan 030051, China

<sup>2</sup> National Key Laboratory for Electronic Measurement Technology, TaiYuan, Shanxi 030051, China

Correspondence should be addressed to Wenyi Liu; [nucliuwenyi@163.com](mailto:nucliuwenyi@163.com)

Received 18 July 2013; Revised 18 December 2013; Accepted 23 December 2013; Published 25 February 2014

Academic Editor: Frank Ehlers

Copyright © 2014 ChengFang Zhen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Both energy-saving and synchronization issues are the paramount concern in wireless sensor networks (WSNs). In this paper we propose a simple and efficient WSN node design based on acoustic positioning applications and present an on-demand sleep/wake scheduling synchronization protocol. Three aspects are already considered in the design: (a) power controllable; (b) energy efficient; (c) high synchronization accuracy. Our primary goal is to maximize energy saving and to control power supplying according to environments and demands. We establish a model of energy consumption and improve it by the ways of power control and on-demand synchronization. The on-demand synchronization protocols are implemented in sensor nodes and evaluated in a testbed. Analysis and simulation were performed that the proposed protocol has significantly reduced the energy consumption. It is also demonstrated by experiments that the platform is accurate and effective.

## 1. Introduction

The latest developments in distributed computing and micro-electromechanical systems have enabled in the past years the emergence of various wireless sensor networks (WSNs) applications comprising military [1], home automation [2], smart building [3], healthy and medical application [4], vehicle and target tracking [5], and industry domains [6, 7]. The WSNs for detection, localization, and tracking of the acoustic sources are widely applied in many industries including defense, robotics, and security sectors. The advantage of objects' localization by their sound radiation is that it is, in essence, a passive method which needs no collaboration with the object in the localization process. Moreover, the localization object is not aware that it is under observation. This is important in defense and object protection applications of WSNs. Another motivation to use localization by sound is that it can be a supplement to other methods of environmental surveillance when visibility is not good enough or when an object is not optically visible [8]. In this paper, we propose a design and

implementation of a wireless sensor network. Our design mainly focuses on energy saving and clock synchronization of the acoustic localization node.

In general, a WSN consists of a large number of low cost and densely deployed battery-powered sensor nodes with wireless communication, sensing, processing, and storage capabilities [6]. The sensor nodes are equipped with sensing, computing, power, and communication modules to monitor a certain phenomenon such as environmental data or object tracking [8]. An important characteristic of sensor devices is that their battery capacity is very small, much smaller than conventional wireless devices like laptops and even PDAs, thereby making energy conservation one of the most important issues in sensor networks. As the sensor nodes are powered by batteries, it is difficult to replace or recharge because of cost (e.g., cost of battery and labor) or geographic (e.g., difficult or unfriendly terrain) reasons and thus the nodes in WSNs are characterized by limited power, processing, and memory resources, making energy saving a paramount concern in WSNs.

Another great challenge in the design of the WSN node is to find suitable mechanisms for clock synchronization [9] in the application of localization. In wireless sensor networks, the basic operation is data fusion, whereby data from each sensor is agglomerated to get a single meaningful conclusion. Sensor readings and timestamps are passed along so that fusion of such information from various sensors will be added up to a global result. In this case, the energy and space might be waste when the object motion was first spotted to its direction. The fusion of individual sensor readings is possible only by exchanging messages that are timestamped by each sensor's local clock. This mandates the need for a common notion of time among the sensors. Protocols that provide such a common notion of time are clock synchronization protocols. Many clock synchronization protocols have been proposed for WSNs [9–11].

Even though many energy-saving sleeping mechanisms have been proposed by researchers [12–16], they are unsatisfactory for every application. Some methods are too complicated with many assumptions on application scenarios, and it is difficult to implement them in real sensor nodes given the constrained resources. Other methods are proposed for general sensor networks, and it is in fact unnecessary to incorporate all of the functions in several situations. In particular, many WSNs are deployed on a small scale in home automation, intelligent buildings, and patient monitoring. They only need a feasible and easily implemented sleep scheduling mechanism. Additionally, the problem of synchronization accuracy and energy consummation always exists.

In this paper, we propose an energy-efficient, simple, and feasible synchronous node sleep/wake mechanism for small scale wireless sensor networks. The node itself can realize power control to sleep or wake according to the demand of the acoustic signal localization. Sensor nodes are divided into sink nodes and listening nodes. Beacon frame containing sleep command from the sink can be forwarded to sink nodes via forwarding nodes. All the nodes in the network can enter sleep mode about the same time. Through such network synchronization mechanisms, we can realize synchronous sleeping and waking of the entire network. Furthermore, a new power control scheme based on synchronization in the medium access control (MAC) layer is proposed. We call it OSWSP (on-demand sleep/wake synchronization protocol). It operates with the help of synchronization protocol and calculates optimal transmission power according to the signal tracking demand. The synchronization protocol includes a wake-up command that begins to work only if the signal has been detected. This approach can greatly reduce the use of energy and cost of wireless sensor network nodes, extending its life period. The proposed mechanisms are implemented in sensor nodes and are evaluated in a test-bed. The analysis and evaluation based on the experimental results confirm that the proposed energy-saving mechanisms are feasible and efficient.

This paper is organized as follows.

In Section 2 previous related works on various energy-saving mechanisms and synchronization protocol based on

existing sensor nodes are summarized. Section 3 introduces the design and implementation of the sensor nodes we proposed. Section 4 presents the synchronous sleep and wake schemes for WSNs and describes the operation of each sensor node. Section 5 describes power control scheme and energy efficiency for WSNs. Section 6 shows the experimental results and analysis. Finally, Section 7 concludes the paper.

## 2. Related Work

The sensor nodes are the basic elements of WSNs, which are responsible for collecting information, fusion processing, and data transmission. Some WSNs were developed for general applications in decades as follows.

At Berkeley, the Smart Dust project [17] aims at developing sensor nodes of micrometric size. They focus on miniaturization of sensor nodes so that it has the size of a dust particle. Since this is a long-time project, the first step was the development of Mote's family. The WeC Mote was one of the first types of sensor nodes developed in this project. Then, they upgraded to Mica Mote and finally to Mica2 Mote. The designer claims that the advantage of this last mote is its radio, which is more robust comparing to TR1000.

The Pico Radio project [18] at Berkeley Wireless Research Center is another project at Berkeley. The objective is to develop a low-cost and low-power sensor node. Its focus is at the radio hardware, link, and network layer stack. Medusa Mk-2 [19] and iBadge [20] are sensor nodes from UCLA. These sensor nodes use more than one processor and iBadge also includes a bluetooth chip. These devices provide a good solution for gateway.

Pushpin [16] is a sensor node, that is, part of a MIT project. Although the main objective is to develop the sensor node for a portable computer, Pushpin's requirements also meet the wireless sensor network needs. It uses a different approach for communication: infrared. Its operational system, Bertha, is for distributed system.

Presently, a few theoretical wireless sensor network node models for monitoring the environment and locating miners have been put forward, but little attention has been given to the synchronization of sensor nodes and energy saving.

The existing literature identifies the idle state of the powered radio of the sensor node to be a major source of power depletion in WSNs [14]. In some applications there are instances in which the traffic load is very light most of the time. It is therefore desirable to turn off the radio when a node is not actively delivering data. There are many approaches to achieving it, such as scheduling the nodes' sleep pattern [12, 13] and imposing control on the radio frequency (RF) radio [15]. Furthermore, Subramanian and Fekri, in their study, try to find the bounds on the lifetime and the sleeping probability of nodes independently, which are valid both for event occurrence detection and continuous monitoring and sensing of an environment [16]. In [21], Anycast packet-forwarding scheme is proposed, where each node has multiple next-hop relaying nodes in a candidate set referred to as forwarding set. Thus, when a node has data to

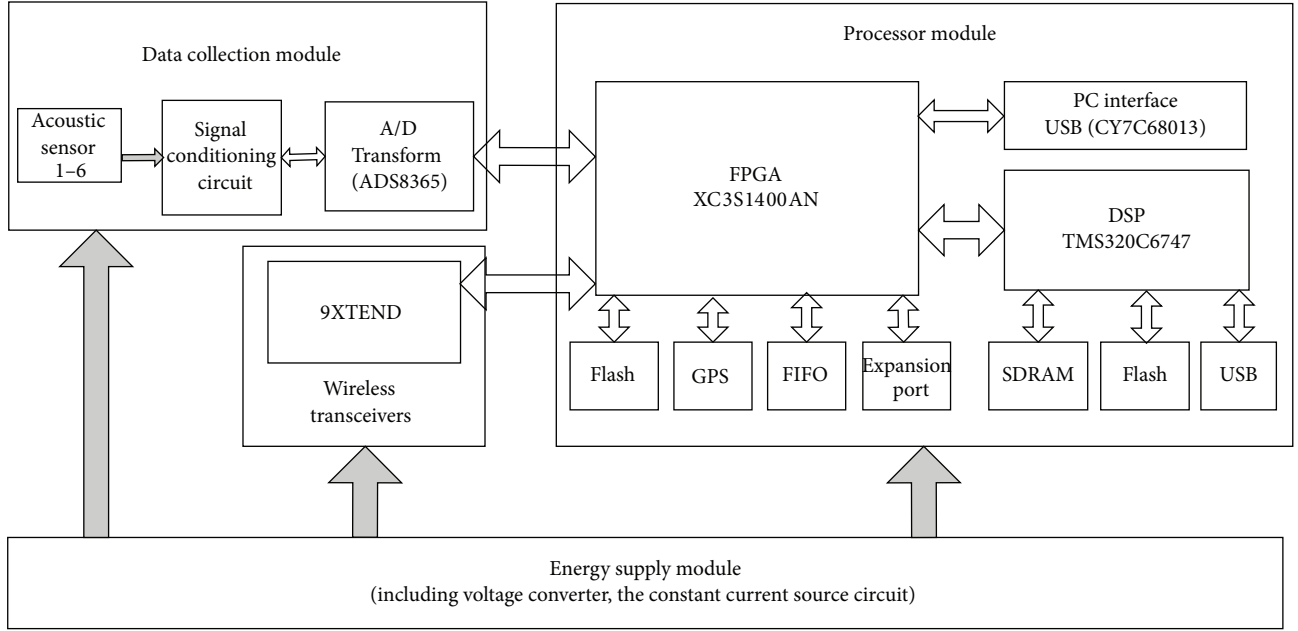


FIGURE 1: The architecture of a sensor node.

send, it has to wait for one nearest specified sink node to wake it up, and then it forwards the packet to the first node that wakes up in the forwarding set. It reduces the expected one-hop delay. In this paper, we compare the proposed protocol with the energy-saving mechanism in [13, 21].

Over the last decade, researchers have developed various clock synchronization protocols for wireless sensor networks which can be classified into two categories: continuous synchronization [10] and on-demand synchronization [11]. In continuous synchronization, time synchronization protocol keeps the clock synchronized at all times so that it can be consulted whenever an event occurs. Among other problems, this strategy wastes energy: if events occur infrequently, the resources that are used to maintain synchronization during idle periods are wasted. In acoustic application, the idle time is longer than the object running time in most cases, so the on-demand synchronization protocols are suitable. Through network synchronization mechanism, we can realize synchronous sleep and wake throughout the entire network.

Having studied previous work on node design, energy-saving sleep, and synchronization algorithm, our work is based on the starting point of these three integrated solutions. The implementation of dormancy mechanisms and synchronization stamp collection are taken into account for the node design, and efforts were made to explore methods for node energy consumption and the accuracy of synchronization. In particular, we have implemented all of the functions described in this paper in real sensor nodes and set up a test-bed to evaluate the proposed mechanisms. Through the analysis of experimental data, it is confirmed that the proposed energy-saving mechanisms are feasible and effective. They can significantly extend the lifetime of network and improve overall network performance.

### 3. Implementation of WSN Acoustic Localization Node (Rewhitened)

**3.1. System Architecture.** WSN nodes are responsible for information collection, calculation and judgment, data processing, and data exchange between neighboring nodes. The typical requirements for WSNs are as follows [1, 8]: reliability, low-power consumption, real-time processing, mobility, accuracy, time synchronization, small size, and low cost. Our node is designed according to these guidelines.

This paper designed a WSN node focusing on the acoustic localization system which can be used for signals detection. Thus, the abnormal status and target movement can be sensed. The node is designed in modular mode. A sensor node is made up of four basic components as shown in Figure 1: a sensing unit, a processing unit, a communication unit, and a power unit. Commensurating with the scalability and flexibility, we deployed some function facilities on the node such as FLASH and FIFO (first in first out) for data storage, USB (universal serial bus) for interface, and redundancy expansion port for expansion.

**3.2. Processing Module.** Proper processor is critical in the design. Most of the typical sensor data processing modules are using a single chip or ARM (advanced reduced instruction set computing machines) chip as a data processor such as AVR microcontrollers of the ATMEL company, ultralow power MSP430 series processors of  $T_1$ , and ARM processors. Though these processors have excellent processing ability, they are not suitable for the application characteristics of acoustic positioning and synchronization data processing needs; its digital processing speed cannot transcend specific DSP (digital signal processor). So we use DSP as data

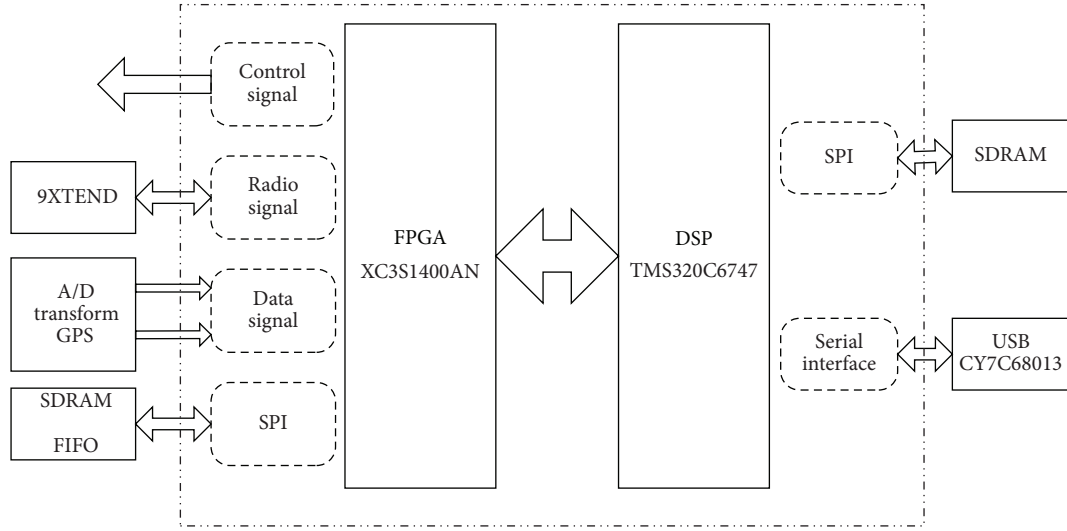


FIGURE 2: The structure of the data processing module.

processing module microprocessors to process the position data and run synchronization algorithm. The FPGA (field programmable gate array) is especially used to collect the timestamp and send the clock data for DSP. This combination makes the sensor more efficient, as shown in Figure 2.

**3.3. Power Supply Control Module.** Power supply module is the cornerstone of the entire node acquisition system, which provides energy for all components. In this design, we specifically designed a voltage conversion circuit and a constant current source circuit to meet different reference voltage needs of the rest of the modules that will use different types of chips ensuring that each chip works in the best possible conditions. In addition, the adoption of the plugged off power chip enabled us to control its power in each module according to the energy-saving requirements.

The power requirements of the sensor nodes are different according to different chips as shown in Table 1.

We use 24 V voltage source series of two 12 V mobile power as the power supply device of the system. The LT1084 voltage converter switches 24 V into 5 V and via the voltage switcher supplies it to other circuits except the sensor, which needed 24 V constant voltage source direct supply.

Specific power supply module structure design schematics are shown in Figure 3.

We ensure that each module can be individually controlled by the power switch in the circuit implementation process, which makes it easy to maximize energy savings. Specific design schematics are shown in Figure 4.

**3.4. Wireless Communication Module.** We use The 9XTend OEM RF wireless chip from DIGI Company for its reliability, high data throughput, long distance ability, and especially for its various switching status shown in Figure 5. 9XTend wireless module is supporting 2.8 V–5.5 V power supply, supporting software programming pin serial port and the

TABLE 1: Voltage levels type involved in system.

Modules	Type of chip	Power requirements
Data collection module	Sensor MPA416	Constant current source, 2.5 V reference
	ADS8365	Analog 5 V, digital 3.3 V
Data processing module	DSP, FPGA	Digital 3.3 V, 1.2 V
	DSP_USB spared	Digital 1.8 V
	SDRAM, FLASH, GPS, USB68013,	Digital 3.3 V
	FIFO	Analog 5 V
Wireless module	9XTEND	Analog 5 V
Others	LED	Analog 5 V

cycle sleep mode, and supporting hardware sleep mode that consumes only 5 mA.

Wireless module supports both hardware off and software off sleep mode, which can achieve low-power design. When the SHDN pin of 9XTend is set low, the wireless module turns into the hardware shutdown mode. The VCC pin current is only 5 uA in this mode. The wireless module will be reset when the SHDN pin is set high, and then after about 100 ms it enters idle mode state. 9XTend can freely be set to sleep mode, serial port sleep mode and cycling sleep mode by using the SM (sleep mode) command. The consumptions of these three modes are 147 uA, 10 mA, and 1.6 mA. The consumptions of the transmission and receiver modes are determined by the data scale and the network congestion state. The consumptions in the different models are shown in Table 2.

**3.5. Sensor Module.** Based on the research background, our targeting object is located in audible voice by human ears. The perception of sound by human ears has a large dynamic range:

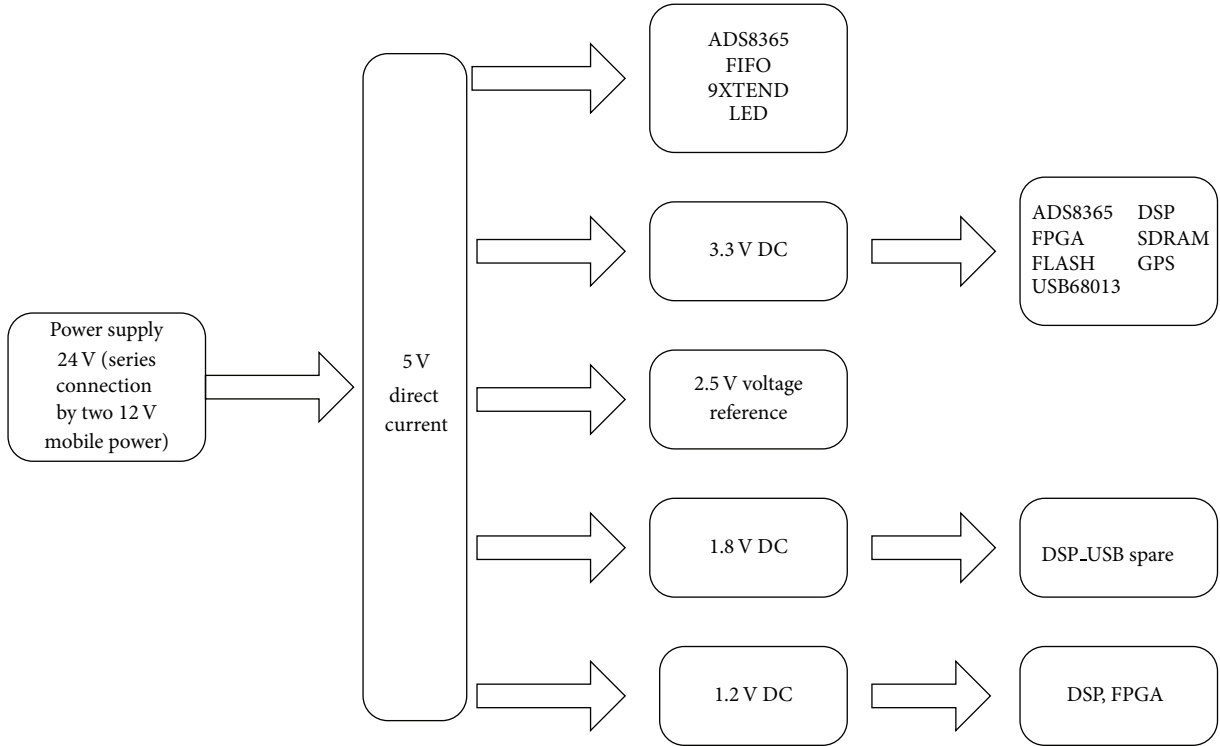


FIGURE 3: Power supplying schematics.

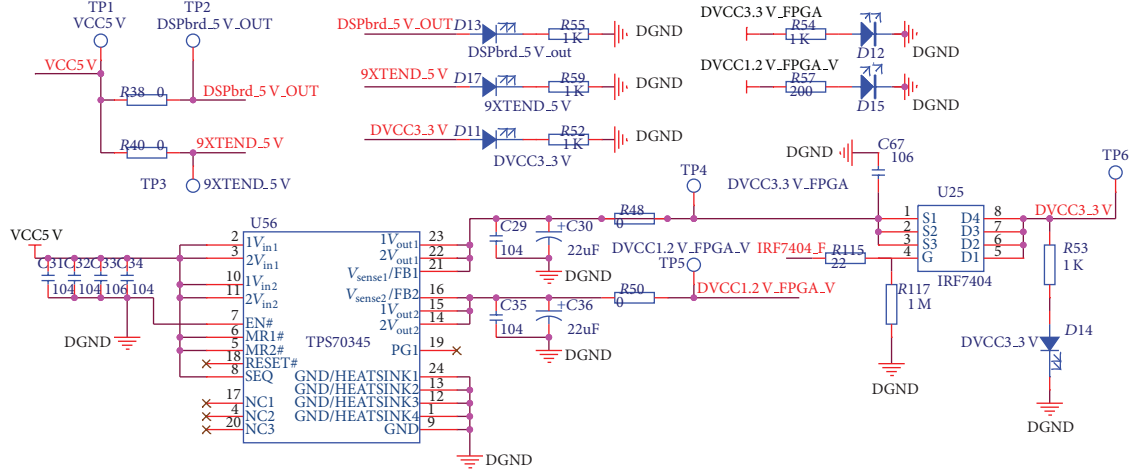


FIGURE 4: Power control circuit principle.

the ratio of permanent hearing-impairing sound in short-time exposure to the silent audible sound is  $10^{12}$ . Such a large range can be expressed in logarithm: the log of  $10^{12}$  at 10 is 12, expressed as 120 db. Human hearing does not have the same sensitivity to all frequencies. The most sensible frequency range is 2–4 kHz. So a microphone of a frequency range of 20 Hz~20 KHz is adopted in our design which carried a terminal modulating circuit.

In acoustics, sound intensity is defined as the power of sound per unit area. In calculating the decibel value, 20 uPa

is the reference value, which is the lower limit threshold in perception of sound by humans. Sound pressure is a field quantity and therefore the decibel is determined by sound pressure. The output voltage of the microphone signal can be calculated by the following sound pressure level formula:

$$Y = 20 \log_{10} \left( \frac{x \text{ (Pa)}}{P_{\text{ref}}} \right) \text{ dB}, \quad (1)$$



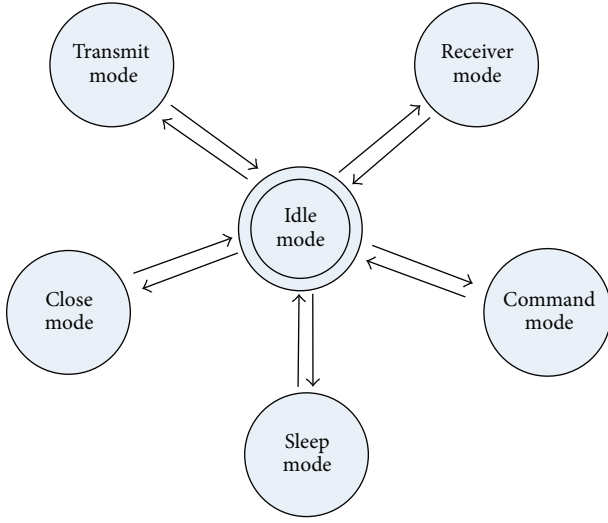


FIGURE 5: The model state transition diagram of 9XTend.

TABLE 2: The consumption of 9XTend in the different modes.

Type of mode	Power value
Off mode	5 uA
Idle mode	147 uA
Serial port sleep mode	10 mA
Cycling sleep mode	1.6 mA
Transmission mode	Decided by data scale
Receiver mode	Decided by data scale

in which  $P_{\text{ref}}$  is the standard reference sound pressure, 20 mPa; we let  $Y$  be 127 dB for experiments command. Then we can get

$$x = 2 \times 10^{Y/20-5} = 2 \times 10^{1.35} \approx 44.77 \text{ (Pa)}. \quad (2)$$

The peak voltage of microphone output can be calculated with the values of the sound pressure and the sensitivity of the microphone:

$$U = (\rho \times x) = (50 \times 44.77) \approx 2.24 \text{ (V)}. \quad (3)$$

So the microphone output voltage range is

$$-2.24 \text{ V} \leq U \leq +2.24 \text{ V}. \quad (4)$$

Obviously, the energy consumption of the sensor observed in the study meets the desired results of an energy efficient design.

#### 4. Sleep/Wake Synchronization Mechanism

Generally, the energy conservation issue is dealt with in five different ways:

- (1) efficient scheduling of sensor node's states to alternate between sleep and active modes;
- (2) efficient control of transmission power to ensure an optimal tradeoff between energy consumption and connectivity;

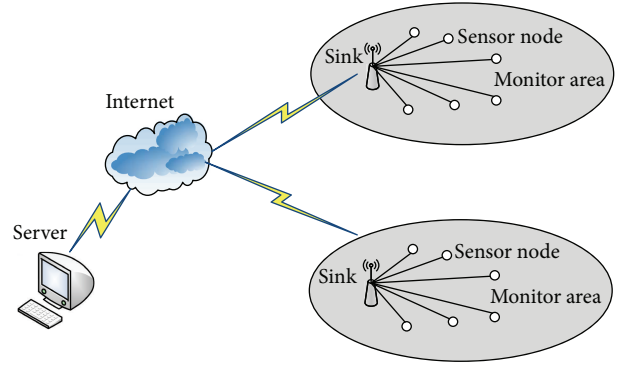


FIGURE 6: System architecture of a WSN.

- (3) energy-efficient routing, clustering, and data aggregation;
- (4) data compression (source coding) to reduce the amount of uselessly transmitted data;
- (5) efficient channel access and packet retransmission protocols on the data link layer.

Here, we mainly focus on the mechanisms on the first and second levels, that is, node sleep scheduling and power control. We name the synchronization protocol we propose as OSSWP (on-demand sleep/wake synchronization protocol).

**4.1. The System Architecture of WSN.** The energy controlling in OSSP is based on the following WSN topology shown in Figure 6. It consists of sensor nodes, a sink node (coordinator), and a server. The sensor node collects environmental information according to the type of sensor used and sends the information to the sink node which may be located multiple hops away. The sink node is also a gateway forwarding data to the server via carrier networks. During forwarding, other intermediate sensor nodes may process the data. The sink node can be an enhanced sensor node or a special network gateway with no sensing capability. It is used to connect a WSN to an exterior network to complete data delivery between the two types of networks. It may be an ethernet gateway if a wired network exists, or it may be a wireless gateway connected to a public cellular network that can significantly extend the WSNs' communication range. A server will receive data, and, in some situations, it will perform some computations and send commands back to sensor nodes via the sink node.

**4.2. On-Demand Synchronous Sleep and Wake Schemes.** Before the sleep and wake schemes are introduced, we briefly describe the beacon frame structure, used to send sleep commands to the entire network. In the IEEE 802.15.4 standard, four types of frames are defined. Data frame is used for the transfer of data. Acknowledgment (ACK) frame and MAC command frame are used for confirming successful frame reception and handling all MAC peer entity control transfers, respectively. Finally, beacon frame is used by the sink to transmit beacons. In our implementation, the beacon

TABLE 3: Beacon frame structure as defined by IEEE 802.15.4.

Octets: 2	1	4/10	2	Variable	Variable	Variable	2
Frame control	Sequence number	Addressing fields	Super-frame specification	GTS	Pending addressing fields	Beacon payload	FCS
	MGR		MAC payload			MFR	

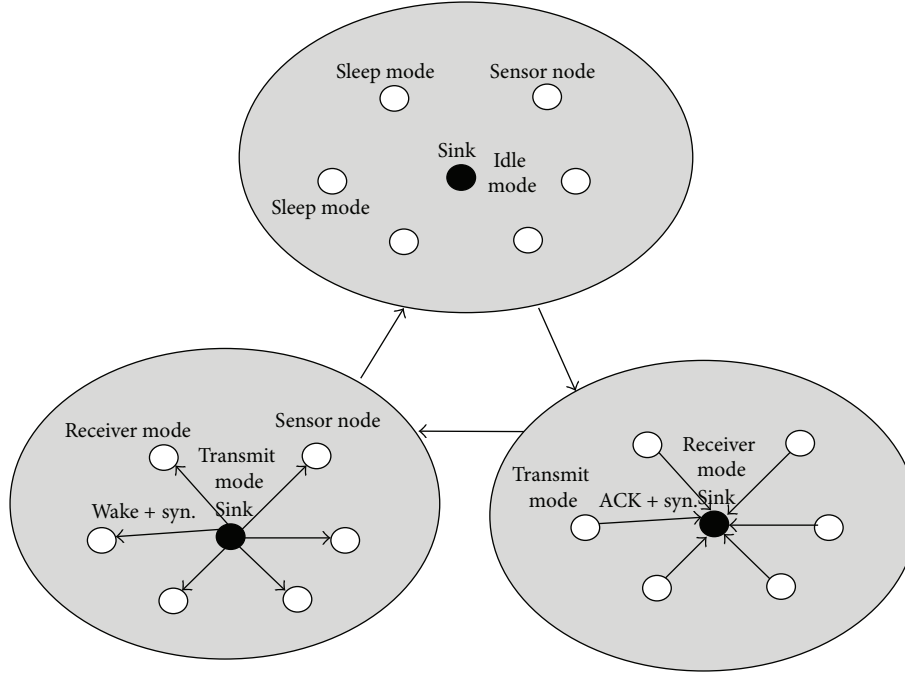


FIGURE 7: WSN mode conversion.

frame is used to implement network synchronization and its structure is shown in Table 3.

In Table 3, MAC header (MHR), MAC payload, and MAC footer (MFR) together form the beacon frame. The MAC payload contains three nonpayload fields (super frame specification, guaranteed time slots (GTS) for quality of service (QoS), and pending address list) followed by the beacon payload field. The MAC payload is prefixed by MHR and appended with MFR. The MHR contains the MAC frame control field, beacon sequence number (BSN), and addressing information fields. The MFR contains a 16-bit frame check sequence (FCS). In this paper, we set “beacon payload” to “01,” which represents sleep command and can be sent by the sink to the entire network for synchronous sleep.

As depicted in Table 3, the sleep command is sent by the sink labeled as “coor” in the figure using the beacon frame. The cluster receives the sleep command from the sink when there is no suspicious target, but the sink is always in the idle mode to monitor the surrounding environment as a sentinel. The synchronization protocol and laicization algorithm does not consume the extra energy but becomes effective only when an interested object event occurs. Once the wireless transceiver of the sink detects the object at the monitor range, it will wake the data processing mode and send a synchronization frame with an awaking command to the nodes in this cluster. All the clusters are thereby turned to the

wake mode from sleep and also running the synchronization protocol to prepare the position of the object. The technical detail of the synchronization protocol is described in other papers by the author, and only the energy-efficiency and basic principles are shown here.

After the sink sends the beacon frame, it enters the receiver mode immediately. In order to ensure that each node successfully receives the beacon frame, a node that has received the beacon frame sends back an acknowledgement frame. Note that the entire network needs to guarantee proper time synchronization when implementing the sleep mechanism. Otherwise, it will cause loss of synchronization among the nodes and will not complete the sleep procedure correctly. The processing of the mode conversion is shown in the Figure 7.

The flowcharts of the sink and sensor are presented in Figures 8 and 9 for better understanding.

Two problems arise from this basic scheme. One problem is that one node may receive multiple sleep commands contained in different beacon frames. The solution is to shut down the receiving function of the wireless communication module after a node has received a beacon frame. The second problem is that it may cause broadcast storm if we let each node rebroadcast the received beacon frame. This problem is solved by a more sophisticated scheme.

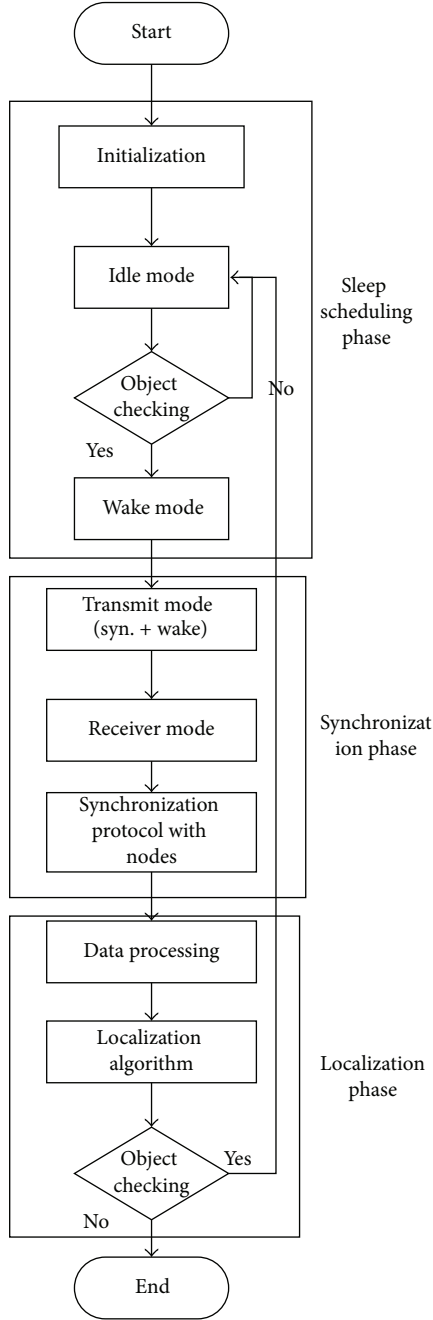


FIGURE 8: Flowchart of major phases and their interaction of the sink.

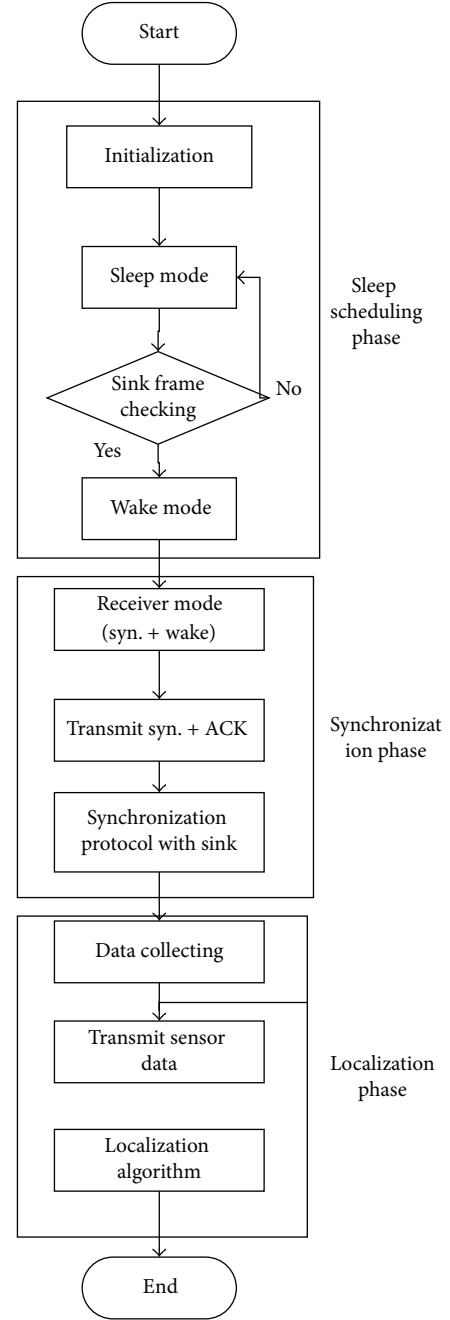


FIGURE 9: Flowchart of major phases and their interaction of the sensor.

**4.3. Enhanced Synchronous Sleep and Wake Schemes.** Considering the problems with the basic sleep scheme, we improve it to optimize network performance by designing a layered architecture of the sensor network. First, we classify the sensor nodes other than the sink into two types: forwarding nodes (FNs) and listening nodes (LNs). Forwarding nodes (FNs) will forward sleep commands from upstream nodes (those closer to the sink) and data from listening nodes; they will not collect environmental information or generate data. Listening node will not forward sleep commands but will collect environmental data and send them out while being

in wake mode. During node initialization, a sensor node is configured as FN or LN. Here, some cluster-based routing protocols are used to select some cluster heads from the sensor nodes. These cluster heads will be chosen as FNs, and the remaining nodes will be designated as LNs. By the way, a FN or LN can also be set manually according to the network topology.

The sink sends out the beacon frame, which contains sleep command information (type = beacon, beacon payload = CMD\_SLEEP). A FN that receives the command will shut down the receiving function of the communication module



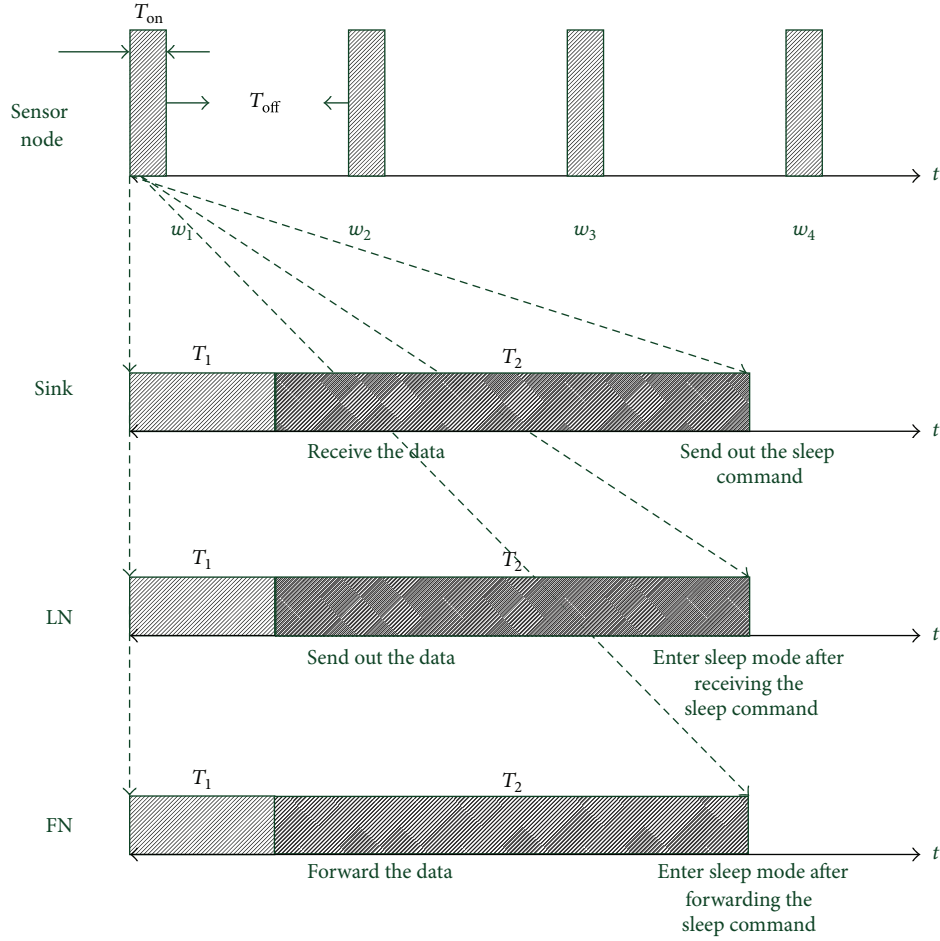


FIGURE 10: Node sleeping and waking scheduling.

and will broadcast the sleep command. It will then enter sleep mode immediately. A LN that receives the sleep command also enters sleep mode immediately. Even though FN and LN enter sleep mode at a slightly different time, the time difference is generally negligible. All of the nodes in sleep will properly set timer counter and wake at about the same time. It is easy to see that the total number of broadcast beacon frames in the network is reduced because LDs will not rebroadcast the received beacon frame.

After waking up, LN will wait for  $T_1$  and send out data. It will make sure that all of its surrounding nodes have been waken. After the interval  $T_1$ , the sink will wait for another interval  $T_2$  during which the next sleep command is sent out; that is, the sink will wait for a total of  $T_0 = T_1 + T_2$  before sending out a sleep command again. The selection of  $T_1$  and  $T_2$  is affected by the accuracy of the system clock and other factors such as network size. In our sensor node, an external crystal oscillator of 8 MHz is used as the system clock with an accuracy of 100 ppm (10<sup>-4</sup>). If the sleep time is set to 900 s, the deviation is  $\pm 0.09$  s and maximal deviation  $\Delta T$  is 0.18 s. Thus, in order to stabilize the system,  $T_1$  should be much larger than  $\Delta T$ . Here we choose 2 s as its value. Considering LN having enough time to send out data that can eventually arrive at the sink via multiple FNs, we set  $T_2$  to 5 s.

Figure 10 shows the timing of the enhanced scheduling scheme for the sleep and wake of sensor nodes. It is assumed that the sensor nodes wake at  $w_1$ ,  $w_2$ , and so on to work for  $T_{on}$  and then sleep for  $T_{off}$  in each cycle. The sink receives the data during  $T_2$  and sends out the sleep command at the end of  $T_2$ . Each LN sends out the data during  $T_2$  and enters sleep mode after receiving the sleep command. Each FN is responsible for forwarding the data during  $T_2$  and enters sleep mode after forwarding the sleep command. Note that different sensor nodes including the sink, LN, and FN enter sleep mode at slightly different times depending on the network size and the accuracy of clock synchronization. Similarly, they also wake at slightly different times. That is why we require that all of the sensor nodes start delivering data after  $T_1$ .

In general, the characteristics and advantages of the proposed synchronous sleep and wake schemes are the following.

- (1) When the sink sends out a sleep command, all of the sensor nodes in the network should be idle because the sink has waited for an extra interval  $T_2$  before sending out the beacon frame. This avoids the loss

of beacon frame that contains the synchronous sleep command.

- (2) The scheme uses beacon frame as carrier of the sleep command which can be processed in MAC layer. Thus, the processing time is short.
- (3) Each node sleeps and wakes at different times. The sensor nodes in the network will not enter sleep mode at exactly the same time. Likewise, they will not wake at the same time either. This can significantly reduce collisions when the sensor nodes have data to send.
- (4) The scheme can decrease the number of beacon frames forwarded in the network and energy consumption.
- (5) After a FN receives a sleep command, it will forward it out and shut down its receiving function. Thus, it will not receive and forward the same sleep command repeatedly. It can further decrease the number of beacon frames forwarded in the network, avoid broadcast storm, and reduce energy consumption.

## 5. Power Control and Energy Analysis

**5.1. Energy Model.** Suppose that the source emits a zero mean random signal  $x_s(t)$  during the time interval of length  $T$ . If the sampling frequency is  $f_s$ , the signal will be represented by  $K = Tf_s$  samples. The measurements of  $x_s(t)$  correspond to the reference distance of 1 m from the source. Under the assumption that the noise is additive, the signal at the  $i$ th sensor node is

$$\tilde{x}_i(t) = x_i(t) + n_i(t), \quad x_i(t) = \frac{x_s(t - \tau_i)}{d_i}, \quad (5)$$

where  $d_i$  is the distance between the source and the  $i$ th sensor node,  $x_i(t)$  is the delayed and attenuated signal at the  $i$ th sensor node,  $n_i(t)$  is white Gaussian noise  $N(0, \sigma_n^2)$ , and  $\tau_i$ ,  $\tau_i = d_i/c$ , is the time delay. The signal energy at the  $i$ th sensor node is modeled by

$$\tilde{E}_i(t) = \frac{E_s(t - \tau_i)}{d_i^2} + \xi_{E_i}(T), \quad (6)$$

where  $\tilde{E}_i(t)$  is the  $K$ -samples based energy estimate defined by

$$\tilde{E}_i(t) = \frac{1}{k} \sum_{l=t-K/2+1}^{t=K/2} \tilde{x}_i^2(l) - \sigma_n^2. \quad (7)$$

And  $E_s(t - \tau_i)$  is the delayed signal energy:

$$E_s(t - \tau_i) = \frac{1}{k} \sum_{l=t-\tau_i-K/2+1}^{t-\tau_i+K/2} x_s^2(l). \quad (8)$$

The estimation error  $\xi_{E_i}(t)$  is

$$\begin{aligned} \xi_{E_i}(t) &= \tilde{E}_i(t) - \frac{E_s(t - \tau_i)}{d_i^2} \\ &= \frac{1}{k} \sum_{l=t-K/2+1}^{t=K/2} \left( (n^2(l) - \sigma_n^2) + 2x_i(l)n(l) \right). \end{aligned} \quad (9)$$

Under the assumption that  $x_s(t)$  and  $n_i(t)$  are uncorrelated, the estimate  $\tilde{E}_i(t)$  is unbiased as  $E(\xi_{E_i}(t)) = 0$ . From (9), the variance of  $\xi_{E_i}(t)$  is

$$\sigma_{\xi_{E_i}}^2 = E(\xi_{E_i}^2(t)) = \frac{2\sigma_n^4}{K} \left( 1 + \frac{2\sigma_s^2}{d_i^2\sigma_n^2} \right), \quad (10)$$

where  $\sigma_s^2, \sigma_s^2 = E(x_s^2(t))$ , is the variance of the signal during source activity.

**5.2. The Influence of  $\log(\det(R_E))$  Term on Estimation of the Acoustic Source Position by Signal Strength.** The covariance matrix  $R_E$  for the signal strength estimates based on  $K$  samples is

$$R_E = \frac{1}{KN} R_{E1}, \quad (11)$$

where  $R_{E1}$  is the covariance matrix for the signal strength estimated by single sample and  $N$  is the order of the matrix  $R_{E1}$ . Substituting (11) into (10), the log-likelihood function becomes

$$L_E(x) = \frac{1}{2} \log(K^N) - \frac{1}{2} \log(\det(R_{E1})) - \frac{1}{2} K^N \xi_E^T R_{E1}^{-1} \xi_E. \quad (12)$$

The first term of (12) is constant and does not influence the estimate of the source position. The second term does not depend on  $K$ . The third term increases with  $K$  according to the factor  $K^N$ . Then, from (12), we can conclude that the second term becomes negligible compared to the third term and can be neglected for sufficiently large  $K$ .

**5.3. The Influence of  $\log(\det(R_T))$  Term on Estimation of the Acoustic Source Position by TOA.** The covariance matrix  $R_T$  is proportional to CRLB constant  $C_\tau$

$$C_\tau R_T = C_\tau^N \bar{R}_T, \quad (13)$$

where  $\bar{R}_T$  is a conditional covariance matrix for the unit value of the CRLB and  $N$  is the order of the matrix  $\bar{R}_T$ . Taking into account (13), the log-likelihood function is

$$L_T(x) = \frac{1}{2} \log(C_\tau^N) - \frac{1}{2} \log(\det(\bar{R}_T)) - \frac{1}{2C_\tau^N} \xi_T^T \bar{R}_T^{-1} \xi_T. \quad (14)$$

The first term does not depend on the source position and can be ignored. For high SNR, the noise variance  $\sigma_n^2$  is small compared to the signal power, and CRLB factor  $C_\tau$

reduces according to (8). The third term of (14) becomes more significant than the second term, which can be neglected. Hence, the likelihood function can be approximated by the third term:

$$L_T(x) \approx -\frac{1}{2} \xi_T^T R_T^{-1} \xi_T. \quad (15)$$

**5.4. Transmission Power Computation.** It is assumed that the distance between two nodes is  $d$ , the power received by the receiving antenna is  $P_r$ , and the output power of the transmitting antenna is  $P_t$ . According to the Friis equation we have

$$P_r = \frac{P_t G_t G_r}{L} \left( \frac{\lambda}{4\pi d} \right)^n, \quad (16)$$

where  $G_t$  and  $G_r$  are the antenna gains of the transmitting and receiving antennas, respectively,  $\lambda$  is the wavelength, and  $L$  is a system loss factor not related to the propagation ( $L \geq 1$ ).

In the free space,  $n$  can be set to 2. In urban situations where there are strong multipath fading effects and there is frequently no clear line-of-sight,  $n$  has to be determined experimentally and is typically in the range of 3 to 5.

We assume that  $P_{thr}$  is the minimal power of received signal that can be decoded correctly. It is easy to determine the minimal transmission power  $P_{min}$  of the sending node to guarantee that the signal can arrive at the receiving node and be decoded correctly.  $P_{min}$  can be computed by the following equation:

$$P_{min} = \frac{LP_t}{G_t G_r} \left( \frac{4\pi d}{\lambda} \right)^n. \quad (17)$$

From (16) and (17), we get the following equation:

$$P_{min} = \frac{P_{thr} P_t}{P_r}. \quad (18)$$

If a receiving sensor node knows the transmission power  $P_t$  of the sending node, the power threshold  $P_{thr}$ , and the received signal strength  $P_r$ , then it can figure out the minimal power that the sending node should use to ensure that the signal can be received correctly. In the following, we call this minimal required power the “optimal power.”

## 6. Simulation and Experiments Results

### Analysis of the Location Node System

The above described synchronization sleep/wake mechanism was formulated in a large scale sensor network of acoustic positioning system which was tested by simulation using OPNET tools. The energy efficiency of our nodes was tested at the National Key Laboratory for Electronic Measurement Technology. The results of formulation and experiments are explained in detail in this section.

**6.1. Simulation of the Synchronization Sleep/Wake Mechanism.** Based on the developed system model, simulations are carried out using OPNET to evaluate the performance

TABLE 4: Simulation parameters for sleep/wake mechanism.

Simulation parameter	Value
Sensing area	$200 \times 200 \text{ m}^2$
Bandwidth	38.4 kbps
Transmission range	50 m
Receiver mode power	30 mW
Idle mode power	30 mW
Sleep mode power	0.003 mw
Transition time	2.45 ms
Packet size	96 bytes
Time slot size	42 ms
Simulation time	300 s

of the proposed synchronization sleep/wake mechanism. Performance of proposed protocol was compared with the two contemporary protocols: TDSS (a target direction-based sleep scheduling algorithm) [13] and Anycast protocol [16]. The following are the details of the simulation setup and discussion of the results.

Simulations were conducted in the sensing area of  $200 \times 200 \text{ m}^2$  and the number of sensor nodes varied from 20 to 260 for different experiments. Sensor nodes were randomly deployed and the random deployment was achieved by choosing  $(x, y)$  locations based on a uniform distribution. The sink node was fixed at the center of the network. The simulations were conducted with a communication range double their sensing range. The simulation parameters are shown in Table 4.

The performance of OSWSP is compared with the TDSS and Anycast protocols.

Experimental parameters, such as average delay per packet and energy per packet, were used to measure the performance of protocol. The synchronization accuracy is considered and described in another paper by the authors [22].

**6.1.1. Average Delay per Packet.** Delay is referred to as the time span between the packet sent from a sensor node and packet received at the sink node. Delay values were measured by changing the number of sensor nodes from 20 to 260. As shown in Figure 11, the average delay experienced by the proposed OSSP protocol was the least, while Anycast was the second and TDSS had the worst delay time. In OSSP, nodes were given different wake intervals according to their traffic requirement with respect to their position in network, their topological importance, and their proximity from the event. The proposed protocol was able to minimize delay at each hop because nodes did not have to wait long for the wakeup interval of the next hop. As a result, average delay per packet in OSSP was less than Anycast protocol and TDSS. In the Anycast protocol, though the node has multiple next-hop relaying nodes by virtue of Anycast packet-forwarding scheme, which helps to find next hop neighbor in a quick manner, it still does not consider the varied traffic requirement of different nodes. Thus, node wait time

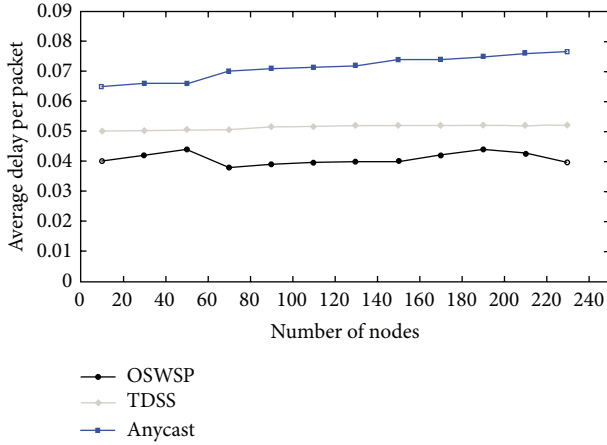


FIGURE 11: Average delay per packet for different numbers of sensor nodes.

increases as the packet approaches to the nodes near to the sink node.

Therefore, it has greater delay than the proposed protocol. In TDSS protocol, nodes have fixed wake interval for the whole network irrespective of their traffic requirement; thus, each node has to wait for the wake interval of the next hop. However, as all nodes have to relay their data all the way to sink node using multihop communication, it involves many relay nodes to reach the sink node, which increases end-to-end delay. Considering the local traffic at each node, sensed data has to wait for some time at each node to get attended; thus delay becomes longer when the packet approaches the nodes near the sink. The problem gets worse when the packet approaches the nodes near the sink node, where the packet suffers maximum delay. It makes the Anycast protocol and TDSS prone to longer delays.

Furthermore, as the number of nodes increases, the OSWSP clearly outperforms the other two strategies. For OSWSP, the performance remains the same for the increased number of nodes, since wake interval is adaptive to the traffic load, whatever may be the size of the network. In this way, increasing the node number has no effect on OSWSP. Therefore, it suggests that the proposed OSWSP protocol is more scalable than TDSS and Anycast protocol.

**6.1.2. Average Energy per Packet.** Average energy per packet is a measure of energy spent for forwarding a packet to the sink node. It is an indicator of the lifetime that can be achieved by the protocols. In Figure 12, average energy per packet is plotted on  $y$ -axis, with varying number of sensor nodes (from 20 to 260) on  $x$ -axis. It can be observed that the average energy consumption per packet for the proposed OSSP protocol is less than Anycast protocol and TDSS, indicating comparatively extended network lifetime. The reason for increased lifetime in OSWSP can be attributed to the fact that it adjusts wake intervals based on traffic loads. By doing so, SMED avoids the case where the nodes remain awake and stay idle when no traffic is to be forwarded. Whereas, in the Anycast protocol, many nodes stay awake to provide alternate

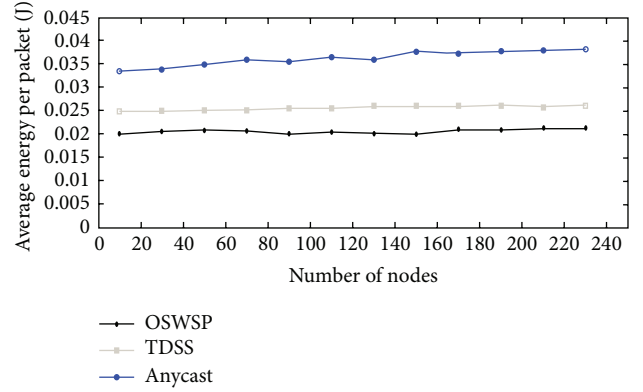


FIGURE 12: Average energy per packet for different numbers of sensor nodes.

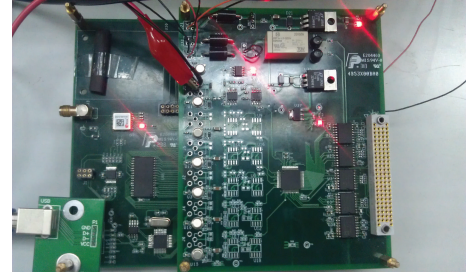


FIGURE 13: The implementation of the sensor node.

paths for routing and mostly they remain idle, as expected, traffic requirements of nodes are considered while setting up sleep/wake schedule. It results in increasing the wake node staying idle, which significantly limits the network lifetime. Similarly in OSWSP protocol random sleep/wake schedule is defined for all the nodes which increases the number of wake idle nodes especially as it moves away from the sink node. Nodes away from the sink node have to do less relaying. Ultimately, it uses the energy of the nodes in idle listening and the lifetime of the network is reduced. Hence proposed protocol has less energy per packet for both Anycast and TDSS protocols.

**6.2. Experiments Results with the Node Power Control.** The proposed power control schemes were implemented in the sensor nodes shown in Figure 13 which were developed at Key Laboratory of Instrumentation Science and Dynamic Measurements. We have set up a test-bed and experiments with sensor nodes developed by the team. The results and analysis are presented in the sequel. In the experiments, two batteries are used (the voltage of each battery is 12 V) in series to provide energy for each sensor node. If the voltage falls below 2.2 V, the sensor node will not work correctly. Thus, we compare the voltage variation in the experiments. Figure 13 shows the variation of average voltage in the sensor nodes before and after the synchronous sleep scheme is used, respectively. Without the scheme, the sensor nodes continuously sense and send information about the



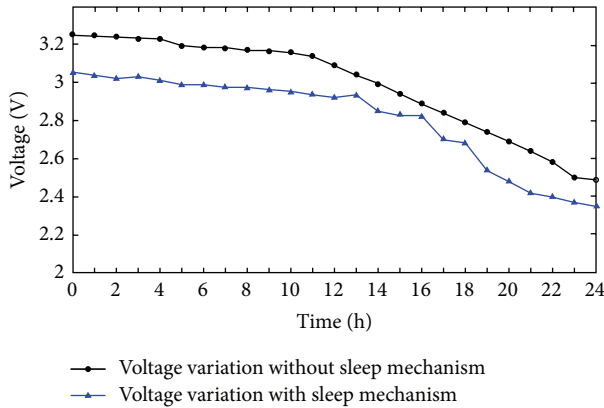


FIGURE 14: Voltage variation analyses before and after sleep mechanism.

surroundings to the server. With the scheme, the node is set freely from sleeping to waking according to the target action.

In our experiment, the sensor was supplied for constant current source in 4 mA, so the voltage variation which is shown in Figure 14 is used to analyze the sensor energy efficiency before and after sleep mechanism.

In Figure 14, the vertical axis represents the average voltage measured in the sensor node. The unit is “hours” and we can see that, within about 22 hours, the average voltage with sleep mechanism is always lower than without the sleep mechanism. The experimental results show that the sensor network will survive for one day depending on whether the sleep mechanism is implemented. It demonstrates that the proposed sleep and wake scheme is effective in saving energy and can significantly extend the lifetime of the network.

Our results show that up to 60% energy savings per battery operated node while maintaining efficient data delivery. The scheduling algorithm allows the nodes to save energy by powering off the wireless communication device. We can use any lightweight timer synchronization mechanism. As demonstrated in the experiments, limiting the number of active nodes in a wireless network not only saves power but also reduces contention achieving higher aggregate throughput. The efficiency of this scheduling technique is proved by the measurements made on a test bed network. In short, the scheduling technique can improve throughput up to 10.3%, with maximum power saving of 85.54%. In terms of energy consumption, a small number of nodes are used for optimal consumption, ensuring that at any point in time the majority of nodes are actually sleeping, resulting in high energy savings (60%).

The above experiment illustrates the operation procedure of the proposed power control schemes implemented in our sensor network. In the following discussion, we compare the voltage variation in the experiments. In Figure 15, it shows the variation of average voltage in the sensor nodes before and after the power control mechanism is used.

In the experiments, the sensor nodes continuously sense and send information about the surroundings such as temperature to the server. From the figure, we can see that, with the proposed power control scheme, the lifetime of sensor

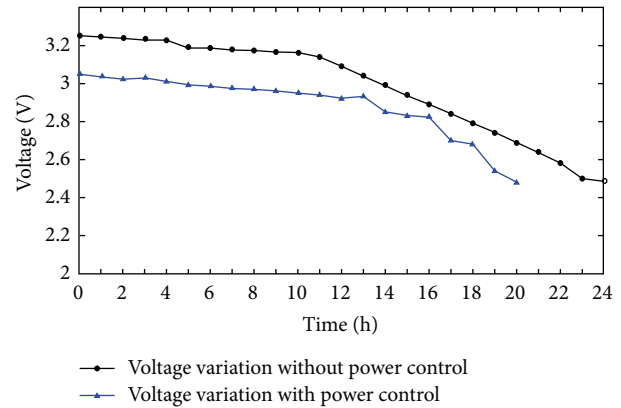


FIGURE 15: Voltage variation with or without power control.

nodes is about 24 hours compared to 20 hours without power control scheme. The lifetime is improved by about 16.7%, which illustrates the efficiency of the implemented power control scheme.

## 7. Conclusions

In this paper, we propose a simple and feasible synchronous sleep and wake mechanism for acoustic positioning WSN. The coordinator of the network will send a beacon frame containing a sleep command to the network and the nodes in the cluster will be responsible for forwarding the command to the attached LNs. Through the network synchronization mechanism, we can realize synchronous sleeping and waking of the entire network. Furthermore, we propose a new hybrid OSWSP. It is located in the MAC layer but is built with the help of microsensor synchronization protocol. The FPGA is used to collect the timestamp of the nodes for the sink, and DSP is used for running the on-demand sleep/wake synchronization protocol according to the tracking of the object. The proposed scheme is easy to implement and does not increase network overhead because it is integrated in the MAC layer. In our study, we have implemented all of the proposed functions in our sensor nodes and set up a real test-bed to evaluate the mechanisms. Through the analysis and evaluation of the experimental results, it is confirmed that the proposed energy-saving mechanisms are feasible and effective.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

The authors wish to thank the editors and reviewers for their valuable suggestions to improve the quality of the paper. This research was supported by the National Natural Science Foundation of China (no. 61071076).



## References

- [1] G. Simon, M. Maroti, A. Lédeczi et al., "Sensor network-based countersniper system," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, pp. 1–12, ACM, New York, NY, USA, 2004.
- [2] M. T. Vo, V. S. Tran, T. D. Nguyen, and H. T. Huynh, "Wireless sensor network for multi-storey building: design and implementation," in *Proceedings of the International Conference on Computing, Management and Telecommunications (ComManTel '13)*, pp. 175–180, Ho Chi Minh City, Vietnam, 2013.
- [3] M. S. Familiar, J. F. Martínez, and L. López, "Smart spaces and environments: a service-oriented middleware architecture for wireless Ad Hoc and sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 725190, 11 pages, 2012.
- [4] M. Al Ameen, J. Liu, and K. Kwak, "Security and privacy issues in wireless sensor networks for healthcare applications," *Journal of medical systems*, vol. 36, no. 1, pp. 93–101, 2012.
- [5] A. Arora, P. Dutta, S. Bapat et al., "A line in the sand: a wireless sensor network for target detection, classification, and tracking," *Computer Networks*, vol. 46, no. 5, pp. 605–634, 2004.
- [6] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.
- [7] A. Flammini, P. Ferrari, D. Marioli, E. Sisinni, and A. Taroni, "Wired and wireless sensor networks for industrial applications," *Microelectronics Journal*, vol. 40, no. 9, pp. 1322–1336, 2009.
- [8] Z. M. Saric, D. D. Kukulj, and N. D. Teslic, "Acoustic source localization in wireless sensor network," *Circuits, Systems, and Signal Processing*, vol. 29, no. 5, pp. 837–856, 2010.
- [9] B. Sundararaman, U. Buy, and A. D. Kshemkalyani, "Clock synchronization for wireless sensor networks: a survey," *Ad Hoc Networks*, vol. 3, no. 3, pp. 281–323, 2005.
- [10] F. Heidarian, J. Schmaltz, and F. Vaandrager, "Analysis of a clock synchronization protocol for wireless sensor networks," *Theoretical Computer Science*, vol. 413, no. 1, pp. 87–105, 2012.
- [11] G. Huang and Y. Albert, "An accurate on-demand time synchronization protocol for wireless sensor networks," *Journal of Parallel and Distributed Computing*, vol. 72, no. 10, pp. 1332–1346, 2012.
- [12] Q. Cao, T. Abdelzaher, T. He et al., "Towards optimal sleep scheduling in sensor networks for rare-event detection," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, IEEE Computer Society, Los Angeles, Calif, USA, April 2005.
- [13] B. Jiang, K. Han, B. Ravindran, and H. Cho, "Energy efficient sleep scheduling based on moving directions in target tracking sensor network," in *Proceedings of the 22nd IEEE International Parallel and Distributed Processing Symposium*, April 2008.
- [14] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '05)*, vol. 3, pp. 1567–1576, IEEE Press, Miami, Fla, USA, March 2005.
- [15] J. Jeong, T. Hwang, T. He et al., "MCTA: target tracking algorithm based on minimal contour in wireless sensor networks," in *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM '07)*, pp. 2371–2375, May 2007.
- [16] R. Subramanian and F. Fekri, "Sleep scheduling and lifetime maximization in sensor networks: Fundamental limits and optimal solutions," in *Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN '06)*, pp. 218–225, April 2006.
- [17] G. P. Forest, "Sizing up smart dust," *Computing in Science and Engineering*, vol. 5, no. 6, pp. 6–9, 2003.
- [18] J. Rabaey and R. Broderickson, "Pico radio: Communication/Computation Pico nodes for Sensor Networks," NTIS Technical Reports Newsletter, 2003.
- [19] <http://research.cens.ucla.edu/projects/2007/Seismic/mexico/wireless/#future>.
- [20] S. Park, "Design of a wearable sensor badge for smart kindergarten," in *Proceedings of the International Symposium on Wearable Computers*, pp. 231–238, October 2002.
- [21] J. Kim, X. Lin, N. B. Shroff, and P. Sinha, "Minimizing delay and maximizing lifetime for wireless sensor networks with anycast," *IEEE/ACM Transactions on Networking*, vol. 18, no. 2, pp. 515–528, 2010.
- [22] C. Zhen and W. Liu, "An accurate on-demand references broadcast synchronization protocol for wireless sensor network," *Sensors and Transducers*, vol. 154, no. 7, 2013.

