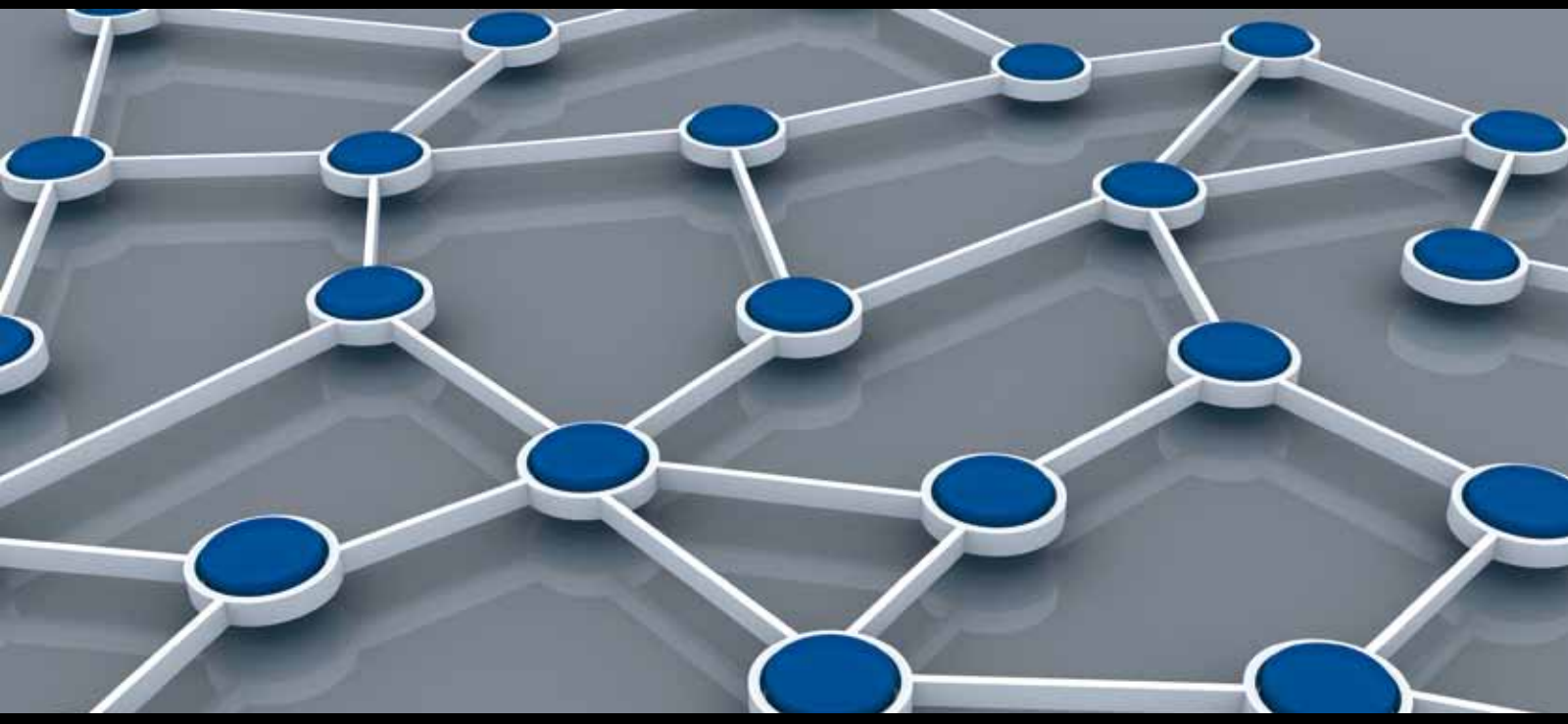
International Journal of Distributed Sensor Networks

Recent Advances in Energy-Efficient Sensor Networks

Guest Editors: Sheikh Iqbal Ahamed, Wei Wang, and Jiman Hong



Recent Advances in Energy-Efficient Sensor Networks

International Journal of Distributed Sensor Networks

Recent Advances in Energy-Efficient Sensor Networks

Guest Editors: Sheikh Iqbal Ahamed, Wei Wang, and Jiman Hong

Editorial Board

Contents

Power Load Distribution for Wireless Sensor and Actuator Networks in Smart Grid Buildings,
Junghoon Lee and Gyung-Leen Park
Volume 2013, Article ID 372982, 8 pages

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 329867, 2 pages
http://dx.doi.org/10.1155/2013/329867

Editorial
Recent Advances in Energy-Efficient Sensor Networks

Sheikh Iqbal Ahamed,[1] Wei Wang,[2] and Jiman Hong[3]

[1] *Department of Mathematics, Statistics and Computer Science, Marquette University, WI, USA*
[2] *Department of Electrical Engineering and Computer Science, South Dakota State University, Brookings, SD, USA*
[3] *School of Computing, Soongsil University, Seoul, Republic of Korea*

Correspondence should be addressed to Jiman Hong; jiman@ssu.ac.kr

Received 7 May 2013; Accepted 7 May 2013

In the past decade, wireless sensor networks (WSNs) have attracted a great deal of research attention, as these networks have the advantage of easy deployment for a wide range of potential applications. Energy efficiency is one of the main goals of WSNs due to that sensor nodes are usually battery powered, and the battery of sensor is not always rechargeable, particularly when the network operates in inhospitable or hostile environments. This special issue on recent advances in energy-efficient WSNs is intended to provide a forum for presenting, exchanging, and discussing the most recent advances in different aspects of the energy efficiency to provide long lifetime for WSNs with emerging technologies.

Reducing energy consumption is a challenge when designing a media access control (MAC) protocol for WSNs. A major source of energy consumption in a WSN is the idle listening mode in which a node remains awake for a long time when no actual data transmission is required by the network. The paper "*Low overhead MAC protocol for low data rate wireless sensor networks*" proposes an energy-efficient, multihop MAC protocol for low data rate sensor networks called LO-MAC. It uses both duty cycling and multihop forwarding from the routing-enhanced MAC protocol (RMAC) to reduce idle listening and sleep latency, respectively. Moreover, LO-MAC exploits common characteristics of wireless communications to build several lightweight energy saving mechanisms.

Multichannel communication protocols have been developed to alleviate the effects of interference and consequently improve the network performance in WSNs requiring high bandwidth. In the paper "*Adaptive learning based scheduling in multichannel protocol for energy-efficient data-gathering wireless sensor networks*," the authors propose a contention-free multichannel protocol to maximize network throughput while ensuring energy-efficient operation. Arguing that routing decisions influence to a large extent the network throughput, they formulate route selection and transmission scheduling as a joint problem and propose a reinforcement-learning-based scheduling algorithm to solve it in a distributed manner.

In WSNs, clustering is widely used to support an effective mechanism in many applications, including environment monitoring, because it promises efficient energy consumption for inexpensive battery-operated sensors. In the paper "*Clustering with one-time setup for reduced energy consumption and prolonged lifetime in wireless sensor networks*," the authors present a novel energy-efficient clustering scheme called clustering with one-time setup which removes the cluster reforming process required at every round after the first round. By removing the cluster reforming process, the number of transmissions per round can be decreased accordingly.

Clustering method for data aggregation in WSNs has attracted great attention for its high efficiency. The paper "*Energy-balanced separating algorithm for cluster-based data aggregation in wireless sensor networks*" focuses on the problem of unbalanced energy dissipation when employing the multihop routing in a cluster-based WSNs. Considering the relaying load undertaken by each cluster, they use the network topology and energy consumption to calculate a cluster radius for obtaining the intercluster energy balancing.

In the paper "*Energy-efficient self-organized clustering with splitting and merging for wireless sensor networks*,"

the authors propose an energy-efficient self-organized clustering model with splitting and merging (EECSM), which performs clustering and then splits and merges clusters for energy-efficient cluster-based routing. It uses information of the energy state of sensor nodes, in order to reduce energy consumption and maintain load balance. In addition, they have shown the validity of splitting and merging of clusters, and then compare the performance of the proposed EECSM with that of a well-known cluster-based self-organization routing protocol for WSNs.

The scarcity of spectrum has become a major bottleneck of the development of the next generation wireless communication system. Cognitive radio (CR), which allows unlicensed or secondary users (SUs) to share the spectrum with licensed or primary users (Pus), shows great promise to enhance the spectrum utilization efficiency. In the paper "*Repeated game-inspired spectrum sharing for clustering cognitive ad hoc networks*," the authors modeled the spectrum sharing problem among multiple SUs as a repeated game. Using the game theory, which has been widely used in designing efficient spectrum sharing, the SUs can iteratively adapt their strategies in terms of requested spectrum size. They analyzed convergence condition under which the total rate revenue of SUs is maximized and the fairness of spectrum sharing. With the proposed clustering procedure and repeated game-inspired model for SUs, a significant performance improvement is achieved compared to other similar spectrum sharing algorithms.

Recently, mobility is an important factor in the design of a routing protocol for WSNs. In the paper "*A mobility-aware efficient routing scheme for mobile sensor networks*," the authors propose a mobility-aware efficient routing, in which sensor nodes make use of mobile information to select the most appropriate routing behavior. The proposed method integrates proactive and reactive routing components efficiently using a sink cluster that consists of underlying multiple static or slow sensor nodes. The cluster provides the stable paths between less mobile entities efficiently.

In the current research inclination, hiring of biological solutions to solve and optimize different aspects of artificial system's problems has shaped into an important field with the name of bioinspired computing. In the paper "*Intelligent optimization of wireless sensor networks through bio-inspired computing: survey and future directions*," the authors have elaborated the importance of bio-inspired algorithms for the optimal solutions of nonbiological systems. The paper leads us to differentiate between various optimization problems existing in WSNs collectively showing that hybrid and non-hybrid algorithms can efficiently optimize the problems in WSNs.

As the developments of various sensor networks, some types of sensors are required to be capable of processing mass data. For example, image sensor nodes take photos using cameras and the images are stored and processed. The paper "*Providing virtual memory support for sensor networks with mass data processing*" deals with supporting virtual memory for sensor nodes which capture and process mass data. Using the optimizing techniques for reducing VM overheads such as cache management, address translation, and secondary storage accessing, the interesting experiments show that data processing using virtual memory can be significantly more energy-efficient than data processing using rich-resource sensor nodes.

The vehicular communication is one of vehicular ad hoc networks (VANETs) research areas using WSNs. In the paper "*An energy-efficient broadcast MAC protocol for hybrid vehicular networks*," the authors have proposed an energy-efficient multihop relay broadcast MAC protocol, which is designed for roadside units downloading service in roadside-vehicular communication systems. The paper focuses on the MAC layer protocol in order to reduce the routing overhead of the frequent topology changes according to a vehicle's movement characteristics. It employs a rebroadcast mechanism based on a vehicle's velocity, distance, and angle from a nearby vehicle for collision-free communication.

Body sensor networks (BSNs) carry heterogeneous traffic types having diverse QoS requirements, such as delay, reliability, and throughput. In the paper "*Traffic priority and load adaptive MAC protocol for QoS provisioning in body sensor networks*," the authors proposed a MAC protocol for QoS provisioning in BSNs and analyzed the effect of prioritizing data packets and dynamically allocating DTS with the consideration of priority and traffic load in the super frame structure of IEEE 802.15.4. The experiments show that the proposed MAC protocol can achieve higher QoS requirements with low power consumption than the state-of-the-art protocols.

The smart grid is a future power system which essentially employs WSNs for its application such as wireless meter reading and remote system monitoring. The power management for actuator gets more important, as many standard protocols such as Zigbee have already achieved significant improvement in sensor network part. The paper "*Power load distribution for wireless sensor and actuator networks in smart grid buildings*" designed an actuator operation scheduler capable of reducing peak load in power consumption of actuator tasks. The most interesting part of the paper is taking advantage of genetic algorithms. Based on the load profile specification and the task model consisting of nonpreemptive and preemptive tasks, each scheduler is encoded into a chromosome, which is an integer-valued vector. The fitness function evaluates the schedule quality by estimating the load of the peaking slot. Using the genetic algorithm, the scheduler achieves acceptable response time with significantly reduced peak load.

Sheikh Iqbal Ahamed
Wei Wang
Jiman Hong

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 217159, 9 pages
http://dx.doi.org/10.1155/2013/217159

Research Article

Low Overhead MAC Protocol for Low Data Rate Wireless Sensor Networks

Kien Nguyen, Yusheng Ji, and Shigeki Yamada

National Institute of Informatics, Department of Informatics (The Graduate University for Advanced Studies), Tokyo 101-8430, Japan

Correspondence should be addressed to Kien Nguyen; kienng@nii.ac.jp

Received 8 November 2012; Revised 29 January 2013; Accepted 24 February 2013

Academic Editor: Jiman Hong

We propose the low overhead media access control protocol (LO-MAC), a new low latency, energy efficient MAC protocol for low data rate wireless sensor networks. LO-MAC uses both duty cycling and multihop forwarding from the routing-enhanced MAC protocol (RMAC) to reduce idle listening and sleep latency, respectively. Besides that, LO-MAC introduces a traffic-adaptive mechanism, which is based on the fact that a node can sense a busy channel within its carrier sensing range. This mechanism uses carrier sensing as a binary signal, and effectively notifies nodes of the existence of a data packet. The nodes then either keep their radios on to take part in multihop data forwarding or turn them off to save energy. Moreover, LO-MAC takes full advantage of the broadcast nature of wireless communication and lets a packet have different meanings when it is in transmission range of different nodes. In LO-MAC, not only is a request-to-send/clear-to-send pair replaced with a Pioneer (PION) packet, as in RMAC, but also a data packet can play both data and acknowledgement functions. Therefore, control overhead and overhearing energy are significantly reduced. Our simulation results show that LO-MAC outperforms RMAC in terms of energy efficiency while achieving comparable end-to-end latency.

1. Introduction

Wireless sensor networks (WSNs) usually contain a large number of wireless battery-attached sensor nodes capable of sensing the environment and communicating in ad-hoc or structure fashion. However, replacing or recharging the batteries used in WSNs is often difficult, so network lifetime may be bound by limited battery capacity [1]. Reducing energy consumption is thus a challenge when designing a media access control (MAC) protocol for WSNs. A major source of energy consumption in a WSN is the idle listening mode in which a node remains awake for a long time when no actual data transmission is required by the network. Several MAC protocol design approaches have been proposed to reduce idle listening energy, the most successful one being MAC in which a duty cycling scheme is applied [2, 3]. In this type of MAC protocol, the nodes periodically alternate between active and sleep states in operational cycles. When being in the active state, the nodes can transmit and receive packets; otherwise, each node turns off its radio.

The original duty cycling scheme, however, adds significant latency in multi-hop packet delivery, since it supports only single hop transmissions. Consequently, a packet that is received at an intermediate node must wait until the upcoming cycle to be forwarded to a neighbor node; the packet hence incurs the so-called sleep latency. The best way to shorten the latency is to increase the number of hops through which the packet can pass in a single cycle. The routing-enhanced MAC protocol (RMAC) [4] introduces a typical solution to reduce excessive end-to-end latency and commences a new class of MAC called multi-hop MAC. The multi-hop MAC preserves the energy-efficient advantage of duty cycling. Moreover, the MAC exploits cross-layer information to relay a data packet via multiple hops in an operational cycle, which contains Sync, Data, and Sleep periods. During the Sync period, the nodes are synchronized their clocks in order to together wake up at the beginning of next cycle. The Data period can be used to schedule multi-hop data transmissions in the subsequent Sleep period. The number of hops a packet can traverse in a cycle mainly

depends on the length of the Data period. The more hops the packet can pass, the longer the required Data period is. By using that long Data period, nodes likely waste energy by keeping their radios on, that is, the long Listening period problem when there is no data transmission in the network.

On the other hand, wireless MAC protocols often use control packets in order to achieve reliability and avoid the well-known hidden/terminal problems, for example, the request-to-send (RTS)/clear-to-send (CTS)/DATA/acknowledge (ACK) handshake in [2, 5]. However, the control overhead, which is the energy consumed to transmit control packets, has been investigated as another large source of energy wastage in WSNs. For example, control packets induces high overheads in the range of 40 to 75% of the channel capacity [6]. Therefore, it is necessary to reduce the control overhead, and intuitively decreasing the number of packets transmitted is only one possible approach. Fortunately, wireless networks are substantially characterized by the broadcast nature, meaning that active nodes usually "overhear" packet content not intended for them when neighboring nodes participate in communication. By exploiting the nature, we can force a packet to play more than one role, as a result both the control overhead and packet overhearing overhead decrease. RMAC also follows this approach and replaces an RTS/CTS pair with a pioneer (PION) packet; however, the protocol still incurs large overhead in data packet transmissions.

We propose the low overhead MAC protocol (LO-MAC), a new duty cycling MAC protocol, which can solve the above-mentioned problems. LO-MAC uses carrier sensing techniques to bypass the long Listening period problem, which occurs very frequently under low traffic load environments. After the Sync period, a short period called the carrier sensing period is introduced. In this period, we use the advantage of the carrier sensing technique to inform the nodes of the status of the network traffic. The nodes then decide to turn their radios off or keep them on to involve in a possible data transmission. LO-MAC is also able to transmit data packets over multiple hops in a single cycle and thoroughly use the broadcast nature to minimize the control overhead. In LO-MAC, not only a control packet replaces an RTS/CTS pair in the Data period, but also a data packet plays both acknowledge and data roles in the Sleep period.

The rest of this paper is organized as follows. Section 2 discusses related work regarding reduced energy consumption and latency MAC protocols. Section 3 gives a detailed description of LO-MAC. Section 4 presents results from the evaluation of LO-MAC. Finally, Section 5 concludes the paper.

2. Related Work

Due to the limitation in battery capacity, energy efficiency is the primary goal in MAC protocol design for WSNs. Initially, many proposed protocols focus on the energy efficiency at the expense of other parameters (e.g., throughput, latency, fairness). Ye et al. [2] have reported that the main source of energy wastage is idle listening energy and propose the sensor-MAC protocol (S-MAC). S-MAC contains the original duty cycling concept to reduce the overhead of idle listening. S-MAC is more energy efficient than the full wake IEEE 802.11 MAC protocol, but it introduces a very large end-to-end latency. Asynchronous MAC protocols such as B-MAC [7], X-MAC [8], RI-MAC [9], A-MAC [10], PW-MAC [11], and CyMAC [12] can free the synchronous overhead but still have the large latency. That is because a WSN is basically multi-hop, and a data packet needs several cycle to reach a multi-hop destination.

Several modified versions of S-MAC have been developed to shorten the latency. One is S-MAC with adaptive listening [3]. In this protocol, a node that "overhears" the control packet (e.g., RTS or CTS) of another node's transmission is going to wake up for a short time when the transmission finishes. If the waken node is the next-hop along a multi-hop path, it and its neighbor immediately receive/transmit the data packet rather than waiting until the next operational cycle. This protocol not only can deliver a packet at up to two hops per operational cycle, but can also significantly increase energy consumption when many neighboring nodes can "overhear" RTS or CTS and wake up, although only one is chosen as the next hop node. Another development is timeout MAC (T-MAC) [13]. T-MAC reduces latency by adaptively changing the ending time of an Active period when there is no traffic transmission near the node. T-MAC can conserve energy better than S-MAC and S-MAC with adaptive listening and can also deliver a packet at most two hops within an operational cycle. In T-MAC, when downstream nodes are unlikely to "overhear" upstream communication two hops away, they are not going to remain awake to receive a forwarded packet. Moreover, T-MAC may also increase energy consumption, since many nodes other than the intended next-hop node are going to remain awake.

Different to the previous approaches, Shu et al. [4] propose RMAC that can forward a data packet via multiple (more than two) hops in a cycle. To enable multi-hop data forwarding, RMAC's nodes relay a cross-layer control packet (i.e., PION) during a Data period in order to schedule a transmission in the subsequent Sleep period. By conveying the cross-layer information, a PION can replace a traditional RTS/CTS pair, hence reducing control overhead. The number of hops over which RMAC can forward a data packet during an operational cycle is limited by the duration of the Data period. That long Data period becomes a source of energy wastage in low data rate environments, that is, the long Listening period problem. Several improvements of RMAC have been introduced [14, 15], but these protocols focused on supporting more data or more hops in a cycle rather than solving the problem. Another development of RMAC is demand wakeup MAC (DW-MAC) [16], which is designed to support dynamic traffic loads. Comparing to RMAC, DW-MAC outperforms RMAC under high traffic loads, but DW-MAC consumes same overhead and achieves higher latency under low traffic loads. That is because in DW-MAC a data transmission starts at a subsequent duration after the beginning of Sleep period which is the starting point of RMAC's data transmission. However, the two protocols share

the manner of relaying control packet during the Data period; hence the problem still remains in DW-MAC.

In our previous work, we have proposed two traffic-adaptive mechanisms to solve the long Listening problem. The first mechanism, which is adopted by MAC^2 [17], uses the first bit of SYNC packet as an indicator of the status of network traffic. The status is then informed to the network by the exchange of SYNC packets. MAC^2 is based on DW-MAC, and outperforms DW-MAC in a wide range of traffic loads. However, under low traffic loads, MAC^2 also introduces longer latency than RMAC with the same reason as in DW-MAC. Being different with the first one, the second mechanism informs the traffic status by exploiting carrier sensing technique. We have applied the second mechanism to RMAC and introduced a new protocol RMAC with carrier sensing (RMAC-CS) [18]. RMAC-CS outperforms RMAC in terms of energy efficiency but has a slightly longer latency and the same control overhead. LO-MAC protocol also adopts the basis of multi-hop MAC and avoids the long Listening problem using carrier sensing technique. The unique characteristic of LO-MAC is that the protocol uses not only a control packet as an RTS/CTS pair, but also a data packet with two data/acknowledge functions. By so doing, LO-MAC can minimize the control overhead. Moreover, our extensive simulations reveal that with a larger value of duty cycle, LO-MAC achieves an even better performance in terms of both energy efficiency and latency under low data rate environments.

3. LO-MAC Descriptions

3.1. Overview. LO-MAC is a duty cycling contention-based MAC protocol, which employs carrier sense multiple access with collision avoidance (CSMA/CA) for accessing channel task. In LO-MAC, the packet structure, short interframe space (SIFS) and distributed interframe space (DIFS) are inherited from IEEE 802.11 [5]. The protocol supports multi-hop transmissions in an operational cycle, which is divided into four periods: Sync, carrier sensing, Data, and Sleep. LO-MAC's nodes wake up together at the beginning of the Sync period, during which they exchange SYNC packets to synchronize their local clocks. In the carrier sensing period, an adaptive mechanism, which exploits the carrier sensing technique, is introduced. The mechanism notifies the nodes of the existence of traffic in the network and lets them keep the radios on or turn them off at the beginning of Data period. In the former case the nodes follow an idle cycle, and in the later case they involve in a busy cycle. In the busy cycle, the nodes may exchange cross-layer control packets during the Data period to schedule multi-hop data transmissions in the subsequent Sleep period.

3.2. Adaptive Mechanism Using Carrier Sensing. As previously explained, the Data period is a key parameter of multi-hop MAC, since it is necessary to initialize multi-hop transmission to achieve a balance between energy efficiency and delivery latency. However, in low data rate environments, nodes may not have packets to be transmitted in most cycles.

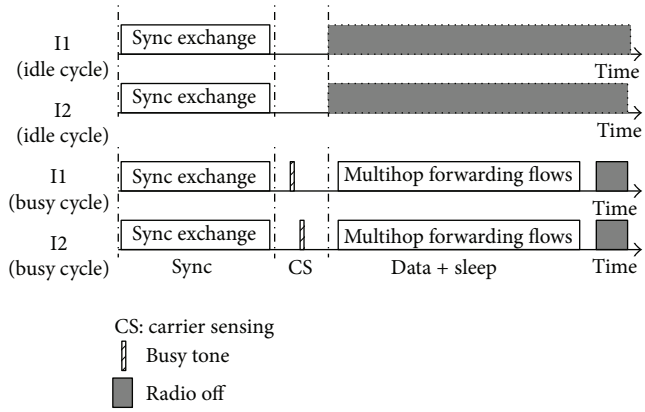


Figure 1: Adaptive mechanism in carrier sensing period.

In this case, nodes needlessly waste energy during the Data period, that is, the long Listening problem. Multi-hop MAC is more efficient if it can let the nodes go to sleep when there is no data in the network; otherwise, it maintains multi-hop flow transmission. To accomplish this, we add a short period called carrier sensing (length T_{cs}) right before the Data period of the multi-hop MAC protocol. During the carrier sensing period, the nodes without pending packets use clear channel assessment (CCA) to determine whether the channel is busy or idle. On the other hand, the nodes with pending data and the node sensing busy channel immediately broadcast busy tones. Thus, all nodes that carrier-sensed the channel as busy or transmitted a busy tone can recognize the existence of data packet and remain on during the Data period. Rather than every node waking up for the entire Data period, a node only wakes up for T_{cs} when there are no packets to be transmitted from its neighborhood nodes.

The benefit of the carrier sensing period is that the total length of the carrier sensing process, even in multiple hops, is negligible if compared with the length of the Data period. This is possible when the CCA time for compliant radio is reported less than 15 μs in a typical sensor mote [19] as well as in the specification of IEEE 802.11 [5]. In our evaluation, we use a much larger T_{cs} which is designed to support a very large number of hops in the multi-hop flow. Additionally, busy tones do not contain any information that needs to be decoded. The only function of busy tones is to enable other nodes to detect the channel as busy. The advantage of not having information in a busy tone is that multiple nodes can transmit simultaneously, causing collisions at the receivers without hindering the protocol. If a collision occurs at a receiving node, the node can still detect the channel as busy and remain on for the Data period. We set the maximum number of busy tones a node can send in the carrier sensing period to one. The nodes those are sending the busy packets cannot sense the channel, but this case still works in LO-MAC because the nodes already know the channel as busy; therefore, they remain awake in the subsequent Data period.

Figure 1 illustrates two types of operational cycles in LO-MAC. Two nodes I1 and I2 exchange SYNC packets during the Sync period. If these two nodes sense the channel as idle

during the carrier sensing period, they follow an idle cycle. Otherwise, they follow a busy cycle and take part in a multi-hop forwarding flow in the upcoming Sleep and Data periods.

3.3. Multihop Data Transmission in a Busy Cycle. In wireless networks, an inherent characteristic is the broadcast nature, meaning that an active node usually "hears" a packet when it is within transmission range of nodes participating in a transmission. Depending on the content of the packet and the process of handling the received packet at the node, the broadcast nature may be advantageous or become a source of overhearing overhead (i.e., the node has to receive useless packets). We use this characteristic to enable multi-hop transmission and conserve energy in designing LO-MAC.

Figure 2 gives an overview of multi-hop transmission in a LO-MAC's busy cycle. The scenario includes four nodes: source S, intermediate nodes (I1 and I2), and destination D. All nodes wake up together at the beginning of the Sync period. They keep their radios on at least until the beginning of the Data period, since the traffic-adaptive mechanism allows them to detect the existence of pending data at S. During the Data period, a multi-hop transmission is initiated as follows. S starts to transmit the first cross-layer control packet to I1, and I1 stores the cross-layer information then modifies and relays the control packet to I2. The process is repeated at I2, and the control packet reaches D. The cross-layer information is used to schedule the wake up time in the subsequent Sleep period. The nodes are woken up at the scheduled time and implement the multi-hop data packet transmission similar to relaying control packet.

3.3.1. Initiating Multihop Transmission. Multi-hop transmission is initiated during the Data period by exchanging the PION packets. In LO-MAC, the construction of a PION packet and the exchange process are inherited from RMAC [4]. The PION packet is constructed by adding cross-layer fields to the original IEEE 802.11 RTS packet. The additional fields, which come from the routing layer, are hop count (the number of hops the PION has traveled) and the final destination. Hence, the new packet PION can play both RTS/CTS roles as well as provide a scheduling function.

A PION is initiated by S and relayed by I1 and I2 during the Data period. During its entire life cycle, a PION packet plays the RTS role regarding a downstream node and the CTS role regarding an upstream node. We describe the PION exchange progress for a 4-node scenario in the Data period shown in Figure 2. The source node has a data packet and starts its PION after a contention window (CW). Intermediate node 1 (I1) receives and relays the PION to its downstream node I2. The PION from I1 serves as both CTS to S and RTS to I2. However, in contrast to the traditional RTS/CTS exchange, the node has to wait at least until the subsequent Sleep period to implement actual data transmission after sending a PION packet. Upon receiving I1's PION, I2 performs the same steps as I1. This process of receiving a PION and immediately transmitting another PION continues until either the final destination has received the PION, or the end of the current Data period has been reached.

On the other hand, the PION packet also provides scheduling information for all nodes in its relay path. The hop count field of a PION packet is used to schedule the wake-up time of nodes in the Sleep period. Unlike in RMAC, the LO-MAC scheduling function works as follows. Suppose that a node is the ith hop during the PION transmission in the current cycle. We denote the node's wake-up time in the upcoming Sleep period as $T_{\text{wake up}}(i)$, which is the subsequent time difference from the start of the Sleep period and calculated as

$$T_{\text{wake up}}(i) = (i-1)\left(l_{\text{DATA}} + l_{\text{SIFS}}\right). \quad (1)$$

Here, l_{DATA} is the time it takes to send a single data packet, and l_{SIFS} is the length of the short interframe space (SIFS) period. To simplify, we assume that all data packets in the sensor network are the same size, so l_{DATA} could be a preset value. Otherwise, the l_{DATA} information can be included in the PION packet, so every node can calculate the correct wake-up time.

3.3.2. Multihop Data Transmission. Similar to RMAC and other multi-hop MAC protocols, LO-MAC's multi-hop data transmission is also implemented in the Sleep period. A source node immediately generates a data packet for a downstream node at the beginning of the Sleep period, and it stays awake for at least the SIFS period plus a small period to receive an acknowledge signal from its neighbor. The data packet needs the SIFS period for packet processing in each node. It is then relayed to a downstream node in the same way as for the PION packet. When the data packet is relayed by each intermediate node, it also plays the role of an ACK packet for the upstream node. If a downstream node fully receives the data packet, it may waste more energy than receiving an ACK. The reason is that the size of data packet is bigger than the size of ACK, for example, four times bigger in RMAC. Therefore the ack function of the data packet is achieved in a different way, as follows.

The node stays in the awake state for a short period ((SIFS + 1) ms in our simulations) after transmitting a data packet to "listen" to the channel. The node verifies the next hop transmission when it receives a peak signal (i.e., the first bit from a transmitting packet). The main idea of the verification is that the node receives only a part of data packet in a similar way as recognizing the Start Frame Delimiter as in [20]. If no packet arrives at the node during the awake time, that means it receives no acknowledge signal. The node then requests for a retransmission of the data packet. If the data packet cannot reach the destination in the current cycle, the last node in the relaying process sends back an ACK packet. On the other hand, when the data packet reaches the destination node, after the SIFS period, the destination node also sends an ACK to confirm the success of flow transmission. The node, which participates in a multi-hop flow and is at one hop before the destination, keeps radio on until it fully receives the ACK packet. In LO-MAC, nodes go

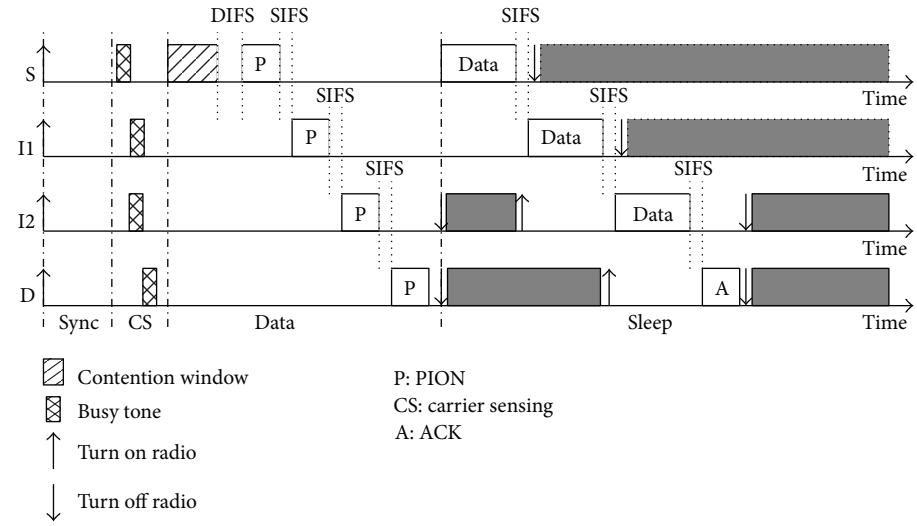


Figure 2: LO-MAC in a busy cycle.

to sleep when they receive either the acknowledge signals or receive the ACK packets.

In the above-mentioned scenario, as soon as the Sleep period starts S and I1 immediately start sending/receiving their data packet. Other nodes in a multi-hop path, which successfully transmits PION in the Data period, go to sleep to save energy. Each node later wakes up at the scheduled time to receive the data packet from the upstream node and relays the packet to the downstream node. For example, node I2 goes to sleep when the Sleep period begins, but it wakes up at the scheduled time when I1 is ready to forward the data packet to I2. This process is described in the Sleep period of Figure 2.

In LO-MAC, we also keep the network allocation vector and frame loss handling as in RMAC.

4. Performance Evaluation

We use the network simulator ns-2 [21] to evaluate LO-MAC's performance. Each sensor node has a single omni-directional antenna through which the combined free space and two-ray ground reflection radio propagation models are employed. Key networking parameters are shown in Table 1, where power consumption parameters are set to typical values for a Mica2 radio (CC1000) [22]. The 250 m transmission range and the 550 m carrier sensing range are modeled after the 914-MHz Lucent WaveLAN DSSS radio interface; although not typical for a sensor node, we use these parameters to make our results comparable to RMAC. In our evaluation of power efficiency, we focus on the energy consumed by radios but ignore the energy consumed by other components such as CPU and memory [23]. The transmission time and size of packets are listed in Table 2. All the duration parameters of an operational cycle of RMAC are presented in Table 3. We denote the length of a cycle and the durations for Sync, Data, and Sleep periods as T_{cycle}, and T_{sync}, T_{data}, T_{sleep}, respectively. These durations are calculated with two different duty cycles at 5 and 10%. Note that the duty cycle parameter

Table 1: Networking parameters.

Bandwidth	20 Kbps
Rx power	0.5 W
Sleep power	0.05 W
Tx power	0.5 W
Idle power	0.45 W
Tx range	250 m
Carrier sensing range	550 m
Contention window (CW)	64 ms
DIFS	10 ms
SIFS	5 ms

Figure 3: Chain scenario.

dc is calculated as follows: $\text{dc} = (T_{\text{sync}} + T_{\text{data}})/T_{\text{cycle}}$, whereas T_{cycle} is fixed at 4465 ms. In the evaluation of LO-MAC, we use the same dc-related and duration-related parameters as in the RMAC's, except the T_{sleep}. Since we add a 5 ms carrier sensing period in LO-MAC, the length of Sleep period is shortened by an amount of T_{cs}.

To simplify our evaluations, we ensure that networks that we use are connected networks. In addition, we do not include routing traffic in the simulations. We also assume that there is a routing protocol deployed to provide the shortest path between any two nodes. We simulate two scenarios: a multi-hop chain and a network scenario. In the chain scenario, the nodes are arranged in a straight line, and their neighbors are placed 200 meters apart, as shown in Figure 3. In the network scenario, 200 sensor nodes are uniformly in a random pattern within a 2000 × 2000 meter square area. The

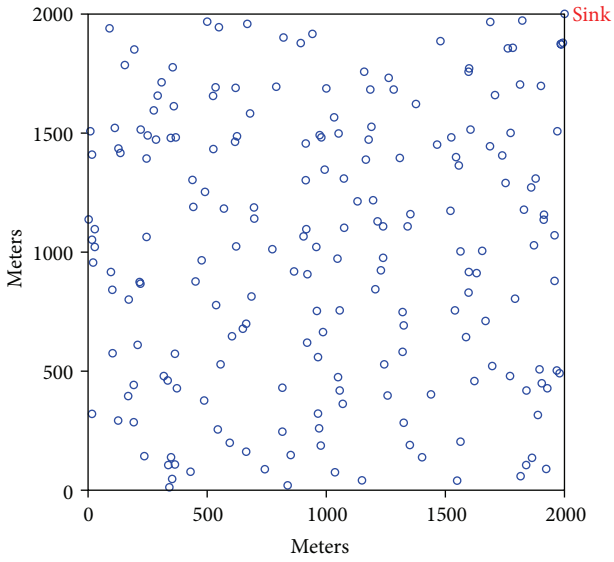


Figure 4: Network scenario: 200 nodes in 2000 × 2000 m area.

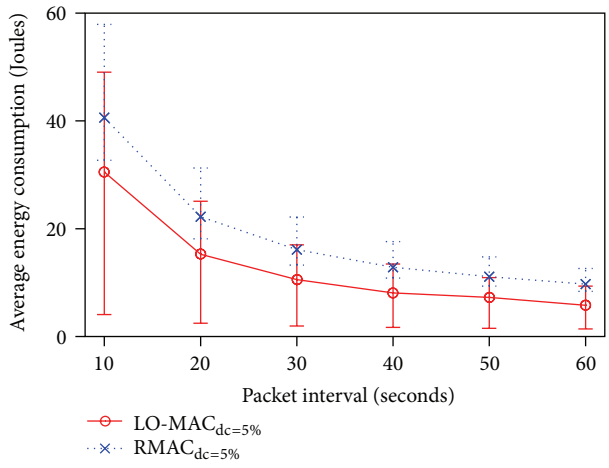


Figure 5: Energy consumption in chain scenario.

sink node is located in the upper right corner of the area, as shown in Figure 4.

4.1. Multihop Chain Scenario. We first evaluate LO-MAC in an 11-node chain scenario. In this evaluation, we use a single flow to send packets at a constant rate from node 0 at the beginning of the chain to the destination node 10, which is farthest node from the source. We vary the interval between two consecutive packets from 10 to 60 seconds. Each simulation lasts for a total of 3600 seconds, and the duty cycle is kept to 5% in all nodes. We compare the performance between LO-MAC and RMAC.

Figure 5 shows the average energy consumption over all the sensors in the chain. The average energy consumption is calculated by dividing the total energy consumed in the simulation by the total number of sensors. The error bars show the minimum and maximum values for a single sensor's energy consumption. When the traffic load increases; that is,

Table 2: Packet parameters.

	Packet size (bytes)	Transmission time (ms)
SYNC/RTS/ACK	10	11
DATA	50	43.0
PION	14	14.2

Table 3: Time duration parameters.

	T_{cycle}	T_{sync}	T_{data}	T_{sleep}
dc = 5%	4465 ms	55.2 ms	168 ms	4241.8 ms
dc = 10%	4465 ms	55.2 ms	391.3 ms	4018.5 ms

the packet interval decreases, the nodes in RMAC and LO-MAC increase their energy consumption, but the LO-MAC's nodes consume less energy than RMAC's nodes. Specifically, when the packet interval is 60 seconds, the average energy consumption in LO-MAC is approximately 75.1% of that of in RMAC, and this value approximates to 59.4% when the packet interval is 10 seconds. We can conclude that LO-MAC outperforms RMAC in terms of energy efficiency. There are two reasons for this. The first one is LO-MAC has the adaptive mechanism; the nodes save power by turning the radio off when no data packet exists in the idle cycles. The second is that during the busy cycles in LO-MAC, the nodes transmit fewer packets than in RMAC. For example, for an N-hop transmission in a cycle, the RMAC's nodes transmit (N − 1) more ACK packets than the LO-MAC nodes do. Moreover, the same amount of energy is consumed for receiving the ACK packets.

Figure 6 shows the average delivery latency in the multi-hop chain scenario. The error bars show the minimum and maximum values of delivery latency. Using an additional period (T_{cs}), the starting point of data transmission (i.e., the beginning of Sleep period) in LO-MAC is later than in RMAC. However, in LO-MAC a node finishes the data transmission sooner than in RMAC, since LO-MAC's node does not send the ACK packets as mentioned above. That means the node just needs a SIFS to start its data relaying. Moreover, the transmission time of an ACK is even longer than T_{cs}, so the delivery latency in LO-MAC is shorter than in RMAC. The difference between those two values is negligible comparing with T_{sleep}. As shown in Figure 6, in this scenario almost the packets need more than one cycle to reach the destination, hence they incur the large sleep latency. In addition, the random process of selecting time slot in the contention window also affects the latency. The sooner the time slot is selected, the better the value of latency is achieved as proven in our previous work [24]. Then we conclude that LO-MAC and RMAC achieve comparable delivery latency in the multi-hop chain scenario.

4.2. Random Network Scenario. In the evaluation of the network scenario, we adopt the same traffic generation method that is introduced in the original RMAC's paper. The traffic load is generated as follows. At a periodic interval of 50 seconds, a sensor node is randomly selected to send one

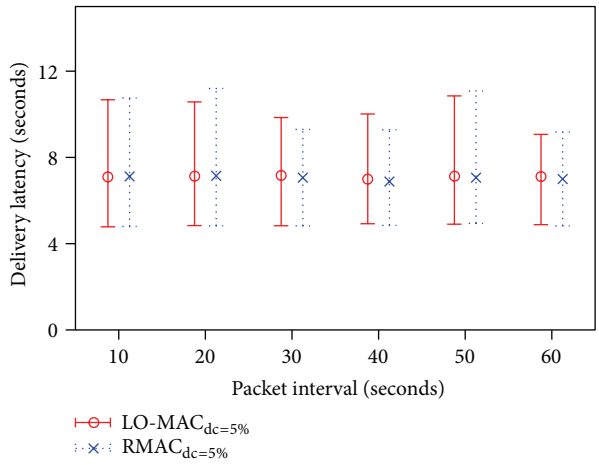


Figure 6: Average latency in chain scenario.

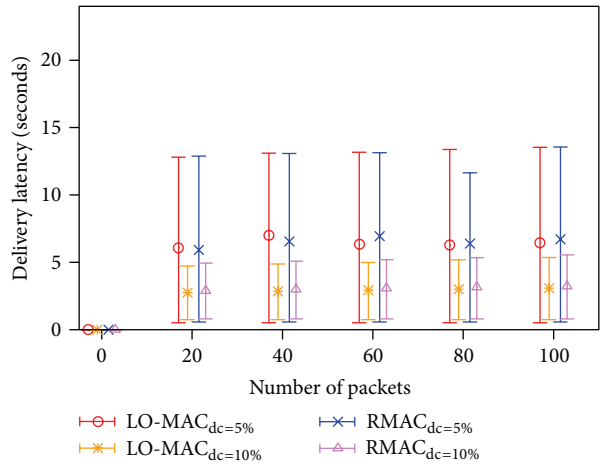


Figure 8: Average latency in network scenario.

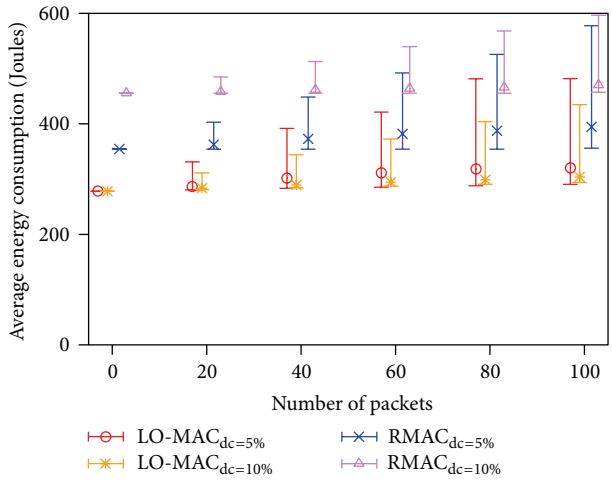


Figure 7: Average energy consumption in network scenario.

data packet to the sink at the top-right corner. If a node is selected to send a packet, it is taken out of the selection pool. The selecting order is similar in both LO-MAC and RMAC's evaluations. The number of packets is varied from 0 to 100; and the total time of each simulation is 5300 seconds. In this evaluation, we use two values of the duty cycle 5% and 10% to investigate the effect of duty cycle to the performance. We denote LO-MAC$_{dc=5\%}$, RMAC$_{dc=5\%}$ and LO-MAC$_{dc=10\%}$, RMAC$_{dc=10\%}$ as LO-MAC and RMAC in the cases of 5% and 10% duty cycle, respectively. The evaluation results for the random network scenario are shown in Figures 7 and 8.

In Figure 7, the middle point is the average value of energy consumption. The average energy consumption value is calculated by dividing the total energy consumption by the total number of nodes. The error bars express the maximum and minimum values of a node's energy consumption during the simulation time. When there is no traffic in the network, the nodes in LO-MAC$_{dc=5\%}$ and LO-MAC$_{dc=10\%}$ consume the same amount of energy, but less than those in RMAC. That shows the maximum effect of the adaptive method in terms of energy saving. In this case, RMAC$_{dc=10\%}$'s nodes consume more energy than RMAC$_{dc=5\%}$'s nodes, since T_{data} in RMAC$_{dc=10\%}$ is larger than the one in RMAC$_{dc=5\%}$. When the traffic load increases, the energy consumption in all four scenarios increases. Among them, LO-MAC$_{dc=10\%}$ achieves the best performance in energy saving. Moreover, the LO-MAC's protocols have better performance comparing with those of RMAC regardless of duty cycle value. That is because 50 seconds is long enough for a packet to be successfully received at the sink, in most cases only one data flow is transmitted in the network. If there is no packet in the network, sensor nodes still consume energy because they have to exchange synchronized information during the Sync period and "listen" to the channel during the carrier sensing period. Another interesting observation from Figure 7 is that with the higher value of duty cycle, the RMAC's nodes consume more energy, but the LO-MAC's nodes do less. That shows the dominant benefit of multi-hop MAC; the longer the Data period is, the more hops the packet can traverse in a cycle. Therefore, a packet may need fewer cycles to reach the destination, then the number of idle cycle is increased, or the more energy is saved.

Figure 8 shows the average value of delivery latency, and the error bar shows the maximum and minimum values. We have the same conclusion as in the chain scenario in the case of 5% duty cycle, since the difference between the average latency in RMAC$_{dc=5\%}$ and LO-MAC$_{dc=5\%}$ is negligible. However, the latency in LO-MAC$_{dc=10\%}$ is slightly shorter than that in RMAC$_{dc=10\%}$. The reason is that all of the packets are transmitted from the random nodes to the sink in one operational cycle.

To furthermore investigate the performance of the protocols, we simulate in the same network scenario with the same traffic model, but the total number of generated packet is 200. The total simulation time is 10300 seconds. We measure the energy consumption of each node and track the delivery latency of all packets. The cumulative distribution function (CDF) of the energy consumption and the delivery latency is shown in Figures 9 and 10, respectively. The results in Figure 9

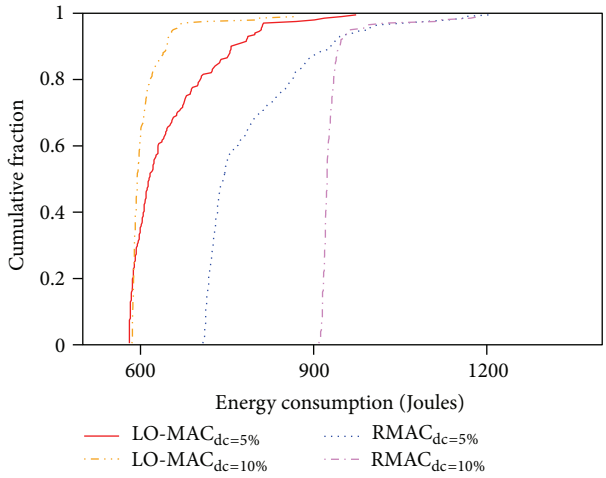


Figure 9: CDF of energy consumption in network scenario.

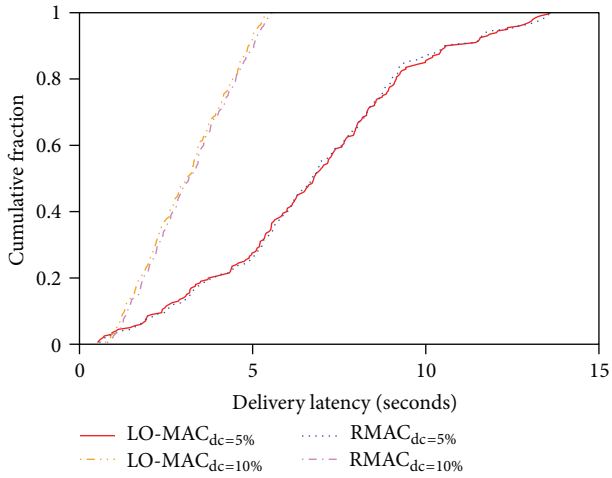


Figure 10: CDF of latency in network scenario.

indicate that even with 5% duty cycle, LO-MAC outperforms the two cases of RMAC, and LO-MAC$_{dc=10\%}$ achieves the best in terms of energy efficiency. Moreover, Figure 10 shows that in LO-MAC$_{dc=10\%}$ the delivery latency is comparable with the one in RMAC$_{dc=10\%}$ but outperforms the others.

5. Conclusion

We propose LO-MAC, which is an energy efficient, multi-hop MAC protocol for low data rate sensor networks. LO-MAC exploits the characteristics of wireless communication to achieve energy efficiency and low delivery latency. The traffic-adaptive mechanism based on carrier sensing effectively controls the period of keeping the nodes' radios on in a cycle, hence, preventing the long Listening period problem. Moreover, LO-MAC relays a packet via multiple hops to reduce end-to-end latency. During the relaying path, a packet from one node often plays two roles to its upstream and downstream neighbors by exploiting the broadcast nature of wireless communication. By doing so, the number of transmissions is significantly reduced; therefore, the protocol can effectively prevent overhearing and control overhead. Our simulation results show that LO-MAC outperforms RMAC in terms of energy efficiency and achieves comparable delivery latency.

References

[1] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, "Energy-aware wireless microsensor networks," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 40–50, 2002.

[2] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the 21st IEEE Conference on Computer and Communications Society (INFOCOM '02)*, vol. 3, pp. 1567–1576, New York, NY, USA, 2002.

[3] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 3, pp. 493–506, 2004.

[4] D. Shu, A. K. Saha, and D. B. Johnson, "RMAC: a routing-enhanced duty-cycle MAC protocol for wireless sensor networks," in *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM '07)*, pp. 1478–1486, Anchorage, Alaska, USA, 2007.

[5] "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," IEEE Standard 802.11, 1999.

[6] A. Woo and D. E. Culler, "A transmission control scheme for media access in sensor networks," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (ACM MOBICOM '01)*, pp. 221–235, July 2001.

[7] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 95–107, November 2004.

[8] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks," in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys '06)*, pp. 307–320, November 2006.

[9] Y. Sun, O. Gurewitz, and D. B. Johnson, "RI-MAC: a receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks," in *Proceedings of the 6th ACM Conference on Embedded Networked Sensor Systems (SenSys '08)*, pp. 1–14, Raleigh, NC, USA, 2008.

[10] P. Dutta, S. Dawson-Haggerty, Y. Chen, C. J. M. Liang, and A. Terzis, "Design and evaluation of a versatile and efficient receiver-initiated link layer for low-power wireless," in *Proceedings of the 8th ACM International Conference on Embedded Networked Sensor Systems (SenSys '10)*, pp. 1–14, November 2010.

[11] L. Tang, Y. Sun, O. Gurewitz, and D. B. Johnson, "PW-MAC: an energy-efficient predictive-wakeup MAC protocol for wireless sensor networks," in *Proceedings of the 30th IEEE Conference on Computer and Communications Society (INFOCOM '11)*, pp. 1305–1313, Shanghai, China, 2011.

[12] Y. Peng, Z. Li, D. Qiao, and W. Zhang, "Delay-bounded MAC with minimal idle listening for sensor networks," in *Proceedings of the 30th IEEE Conference on Computer and Communications Society (INFOCOM '11)*, pp. 1314–1322, Shanghai, China, 2011.

[13] T. van Dam and K. Langendoen, "An adaptive energy efficient and low-latency mac for data gathering in wireless sensor networks," in *Proceedings of the 1st ACM Conference on Embedded Networked Sensor Systems (SenSys '10)*, pp. 171–180, 2003.

[14] K. T. Cho and S. Bahk, "HE-MAC: hop extended MAC protocol for wireless sensor networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '09)*, pp. 1–6, December 2009.

[15] T. Canli and A. Khokhar, "PRMAC: pipelined routing enhanced MAC protocol for wireless sensor networks," in *Proceedings of the IEEE International Conference on Communications (ICC '09)*, pp. 1–5, June 2009.

[16] Y. Sun, S. Du, D. B. Johnson, and O. Gurewitz, "DW-MAC: a low latency, energy efficient demand-wakeup MAC protocol for wireless sensor networks," in *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '08)*, pp. 53–62, May 2008.

[17] K. Nguyen, U. Meis, and Y. Ji, "MAC^2: a multi-hop adaptive MAC protocol with packet concatenation for wireless sensor networks," *IEICE Transactions on Information and Systems*, vol. 95, no. 2, pp. 480–489, 2012.

[18] K. Nguyen and Y. Ji, "Using carrier sensing to improve energy efficiency of MAC protocol in sensor networks," in *Proceedings of the 10th International Symposium on Communications and Information Technologies (ISCIT '10)*, pp. 70–74, October 2010.

[19] J. Zhao and R. Govindan, "Understanding packet delivery performance in dense wireless sensor," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pp. 1–13, November 2003.

[20] Z. Lu, T. Luo, and X. Wang, "XY-MAC: a short preamble MAC with sharpened pauses for wireless sensor networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '12)*, pp. 1550–1554, Paris, France, 2012.

[21] The network simulator—ns2, http://www.isi.edu/nsnam/ns/.

[22] W. Ye, F. Silva, and J. Heidemann, "Ultra-low duty cycle MAC with scheduled channel polling," in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys '06)*, pp. 321–334, November 2006.

[23] V. Shnayder, M. Hempstead, B. R. Chen, G. W. Allen, and M. Welsh, "Simulating the power consumption of large-scale sensor network applications," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 188–200, November 2004.

[24] K. Nguyen and Y. Ji, "Achieving minimum latency in multi-hop mac protocol for wireless sensor networks," in *Proceedings of the IEEE 73rd Vehicular Technology Conference (VTC '11)*, pp. 1–5, May 2011.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 487846, 11 pages
http://dx.doi.org/10.1155/2013/487846

Research Article

Energy-Efficient Self-Organized Clustering with Splitting and Merging for Wireless Sensor Networks

Kyuhong Lee and Heesang Lee

Department of Industrial Engineering, Sungkyunkwan University, Suwon 440-746, Republic of Korea

Correspondence should be addressed to Heesang Lee; leehee@skku.edu

Received 26 October 2012; Revised 24 January 2013; Accepted 20 February 2013

Academic Editor: Wei Wang

Energy efficiency is one of the most important issues for WSNs, because the battery of each wireless sensor node cannot be recharged or replaced. Therefore, all batteries have to be well managed, in order to provide a long network lifetime, as well as to reduce energy consumption for WSNs, particularly in clustering and routing. In this paper, we propose an energy-efficient self-organized clustering model with splitting and merging (EECSM), which performs clustering and then splits and merges clusters for energy-efficient cluster-based routing. EECSM uses information of the energy state of sensor nodes, in order to reduce energy consumption and maintain load balance. We show the validity of splitting and merging of clusters and then compare the performance of the proposed EECSM with that of a well-known cluster-based self-organization routing protocol for WSNs. The results of our experiment show good performance of EECSM, in terms of network lifetime, residual energies, scalability, and robustness.

1. Introduction

A sensor node of a wireless sensor network (WSN) senses the environment, and the information obtained from the environment is converted into data packets and then transmitted to the *base station* (BS), which is the external final processing node for users [1]. According to the advancement in wireless sensor devices, WSNs have performed important roles in recent years, such as environmental monitoring and target tracking.

WSNs have many unique characteristics, as well as the following constraints. First, a WSN consists of hundreds or thousands of wireless sensor nodes. Therefore, each wireless sensor node should consist of several cheap devices, which are constrained in terms of the processing capability and storage capacity. Second, the battery of each wireless sensor node cannot be recharged or replaced, and thus all batteries have to be well managed, in order to provide long network lifetime and reduce energy consumption for WSNs [1, 2].

For the energy-efficient transmission of information obtained by sensor nodes, many routing protocols have been developed. The routing protocol decides the transmission route for a data packet, from the sensor node to the external final destination BS. In general, routing protocols in WSNs can be divided into *flat routing, cluster-based routing,* and *location-based routing*, depending on the network structure. Sensor nodes of flat routing typically have identical roles, whereas sensor nodes of cluster-based routing have different roles, in order to create hierarchy. Meanwhile, location-based routing uses the location of sensor nodes for routing. Among these routing protocols, cluster-based routing protocol is the most energy efficient [3]; therefore, we study cluster-based routing in this paper.

In cluster-based routing, a network consists of several clusters, and each cluster is comprised of a *cluster head* (CH) and many *cluster members* (CMs). Due to the limitation of a wireless sensor node, a centralized clustering and routing method is almost impossible in WSNs.

Considering these limitations, a distributed clustering and routing method is normally used for WSNs. In distributed clustering and routing, each sensor node can reduce energy consumption, by clustering when transmitting the data packets.

Self-organization, which is a further development of distributed processing, has recently been studied intensively. Although wireless sensor nodes have limited capacities,

the self-organization method can perform adequate clustering and routing, by using only localized neighbor information and simple rules. Self-organization processing also has advantages, such as scalability and robustness.

Although the size of clusters should be adjusted properly, in order to maximize energy efficiency, the usual cluster-based routing protocols do not guarantee proper clustering size, since they only use localized neighbor information. Since the CMs send data packets to a distant CH when the size of a cluster is too large, high energy consumption can occur. In addition, the CH may experience transmission delays and cause bottlenecks. In contrast, when the size of clusters is too small, the number of clusters is increased, engendering an increase in the number of CHs, which consume much more energy when compared with CMs.

In this paper, we propose *an energy-efficient self-organized clustering with splitting and merging* (EECSM) model for WSN. In order to increase energy efficiency, EECSM performs clustering, by considering the energy state information and also by deciding on the split or merge clusters. This action is performed by monitoring the size of the clusters in self-organized ways.

This paper is organized as follows. Section 2 reviews the previous studies on cluster-based and self-organized routing and on the protocols that use splitting and merging clusters. Section 3 presents the radio model and the definitions used in this study. Section 4 explains the process of the EECSM model. Section 5 proves the validity of splitting and merging and compares the performance of the proposed EECSM with that of another well-known clustering model. The last section presents the conclusions of this study and proposes future research directions.

2. Related Works

According to the energy state or the location of sensor nodes, sensor nodes have different roles: CH, CM, gateway, and so on. The data transmission of cluster-based routing is performed in accordance with the following procedures. (i) The CMs transmit the data packet, created by the sensing environment, to their CHs. (ii) The CHs aggregate the data packets received from their CMs and transmit them to the BS, either directly or via other CHs [3].

Cluster-based routing protocols have advantages of energy efficiency and scalability, and clustering can localize the transmission of signals that is used to maintain the routing paths and clusters. In addition, localized transmission avoids the redundant exchange of messages among sensor nodes. According to the decrease in the transmission range, clustering can reduce the transmission energy of the data packets [3–6]. A single-tier network (flat routing protocol) can cause sensor nodes to overload with an increase in the number of sensor nodes; it does not offer scalability. Cluster-based routing protocols have more than two tiers; thus, they can prevent an increase of overload. Therefore, cluster-based routing is the most proper method to cover a large area without any service degradation [4].

Recently, research on "Self-organization" has advanced in many fields, such as forming clusters and routing packets in WSNs or ad hoc networks [7, 8]. Here, "self-organization" refers to a network that is organized without any centralized control mechanism, using only local information and simple rules, that is, solely organized through a distributed method. The advantages of using self-organization are completely distributed control, adaptability to a changing environment, robustness of a sudden change of environment, and scalability. Several cluster-based self-organization protocols for WSNs were studied as follows.

The Low-Energy Adaptive Clustering Hierarchy (LEACH) protocol [9] is a well-known self-organized cluster-based protocol. The LEACH decides CHs by a randomized rotation, in order to distribute the energy consumption of sensor nodes. Each sensor node compares the threshold $T(n)$ that has a random number, in order to elect a CH. The CHs are changed periodically, in order to balance the energy of sensor nodes. Sensor nodes of LEACH organize the clusters on their own, by using both local decision and local communication [7, 9].

LEACH-Energy Distance (LEACH-ED) [10] is another self-organized cluster-based protocol. The LEACH-ED decides CHs based on two thresholds. First, the threshold is the ratio between the residual energy of a sensor node, and the total current energy of all of the sensor nodes in the network. Second, the threshold is the distance between some nodes that is less than the distance threshold; only one node becomes a CH. The LEACH-ED improves the load balance and the network lifetime better than does the LEACH.

The Hybrid Energy-Efficient Distributed Clustering Approach (HEED) [11] also uses clustering techniques for energy efficiency. The HEED decides CHs by two parameters. The first parameter is the residual energy of the sensor nodes. The second parameter is the intracluster communication cost to solve break-ties. The intra cluster communication costs are the (i) minimum degree cost, (ii) maximum degree cost, and (iii) average minimum reachability power (AMRP). The HEED prolongs the network lifetime longer than that of the generalized LEACH.

Robust Energy-Efficient Distributed (REED) clustering [12] uses clustering techniques for energy efficiency and robustness from unexpected failures or attacks on a CH. REED achieves fault tolerance by constructing multiple independent CH overlays, on top of the physical network. The CH selection method of REED is similar to that of HEED.

To maintain high energy efficiency, some cluster-based algorithms utilize one of several splitting and merging clustering methods, each of which adjusts the size of the cluster by itself [13–15]. Algorithms for the splitting and merging of clusters of WSNs were studied as follows.

To detect and track continuous objects (e.g., biochemical material), a location-based clustering method [13] is proposed. A boundary sensor detects an object in its vicinity, but has one or more one-hop neighbors that do not detect the object. To reduce the communication cost, the algorithm organizes boundary sensors into several clusters. If a cluster is larger than the predefined maximum cluster size, the cluster is split into two clusters. If a cluster is smaller than

the predefined minimum cluster size, the cluster is merged into another cluster. However, that paper presents only the concept of splitting and merging. The actual process of splitting and merging is not addressed.

The energy-balanced dispatch algorithm [14] focuses on the energy efficiency problem of dispatching mobile sensor nodes to event areas. To reduce the energy consumption of dispatching mobile sensor nodes, the algorithm assigns each dispatching mobile sensor node to each cluster. At the clustering process, the algorithm splits and merges iteratively, to balance clusters and reduce the costs of clusters. However, the splitting and merging are operated not in a distributed way, but by a centralized dispatch algorithm.

The clustering-based data collection algorithm [15] focuses on the energy efficiency problem of clustering and prediction. In clustering, this algorithm uses dynamic splitting and merging clusters, in order to reduce the communication cost. However, these clustering, splitting, and merging methods are operated not for energy-efficient routing, but for using AR model-based similarity of features of CH and CMs.

In this paper, we propose EECSM for energy-efficient clustering of WSN. To reduce the energy consumption of CHs and CMs, EECSM adjusts the size of the clusters. When the size of a cluster is too large, EECSM splits the cluster into two clusters. On the other hand, when the size of a cluster is too small, EECSM merges the cluster into other clusters. Through splitting and merging clusters, EECSM prolongs the network lifetime.

3. Network Modeling

3.1. First Order Radio Model. A sensor node consumes energy, when transmitting and receiving data packets in a WSN. In wireless data transmission, energy consumption is correlated to the data packet size and the distance between the two sensor nodes. Extensive research has been conducted in the area of low-energy radios. Different assumptions about the radio characteristics, including energy dissipation in the transmission and receive modes, will change the advantages of different protocols. In our work, we assume the following first order radio model as the radio energy consumption model [9]. We also assume that a sensor node can identify its own energy, by using the radio model. This assumption is also used in LEACH [9] and HEED [11].

(i) Transmitting the data packet: a sensor node consumes $\varepsilon_{\text{elec}} = 50\,\text{nJ/bit}$ at the transmitter circuitry and $\varepsilon_{\text{amp}} = (100\,\text{pJ/bit})/\text{m}^2$ at the amplifier.

(ii) Receiving the data packet: a sensor node consumes $\varepsilon_{\text{elec}} = 50\,\text{nJ/bit}$ at the receiver circuitry.

(iii) A k-bit data packet is transmitted from sensor node to sensor node, and d_{ij} is the distance between the two sensor nodes i and j: the energy consumption of the sensor node i is given by $T_{ij} = \varepsilon_{\text{elec}} \times k + \varepsilon_{\text{amp}} \times d_{ij}^2 \times k$.

(iv) The sensor node receives the data packet: the energy consumption of the sensor node is given by $R_i = \varepsilon_{\text{elec}} \times k$.

3.2. Definitions for the Model. We define the following terms for our EECSM model.

(i) Period: in the data transmission phase, data packets are transmitted from all CMs to the BS. A cycle of this process is defined as a "period."

(ii) Clustering round: the number of repeated periods during a data transmission phase. EECSM maintains the same cluster configuration of the WSN during one clustering round.

(iii) Network lifetime: the periods until a certain number of sensor nodes (e.g., 30 sensor nodes) are all discharged of their energy.

(iv) Merging threshold: if the number of CMs of a cluster is below the merging threshold, the EECSM merges the cluster into other clusters.

(v) Splitting threshold: if the number of CMs of a cluster is above the splitting threshold, the EECSM splits the cluster into two clusters.

4. Energy-Efficient Self-Organized Clustering with Splitting and Merging (EECSM): The Proposed Clustering-Based Routing Protocol

We propose EECSM for a WSN. To achieve energy efficiency, EECSM has the following five considerations, using the local information. First, sensor nodes that have the most remaining energy from neighbors become the candidates for the CHs, since the CH consumes a lot of energy. Second, each sensor node, except CHs, selects the nearest CH, in order to minimize energy consumption during the data transmission phase. Third, a cluster that has a number of CMs that is less than the merging threshold merges into other clusters, in order to minimize energy consumption of transmitting the data packets from the CHs to the BS. Fourth, a cluster that has a number of CMs that is larger than the splitting threshold splits into two clusters, in order to reduce the overhead of the CH. Fifth, to prevent any breakdown of CHs, a CH-backup mechanism selects a new CH that has maximum energy in the cluster.

The performance of EECSM is focused on clustering, splitting, and merging. For proper performance evaluation, EECSM does not use a particular routing method between CHs. This means that the CHs directly transmit data packets received from their particular CMs to the BS.

EECSM is comprised of three phases and a CH-backup mechanism; the *clustering phase* forms the clusters for the WSN (in Section 4.2); the *merging cluster phase* decides whether to merge clusters or not (in Section 4.3); the *data transmission phase* sends/receives the data packets (in Section 4.4); and the *CH backup mechanism* elects a new CH when the CH is a breakdown (in Section 4.5).

Before explaining the EECSM model, we explain the transition of state for each sensor node.

4.1. States of Sensor Nodes. Each sensor node of EECSM performs its roles while being transformed into the following four states (*undecided state*, *CH state*, *CM state,* and *discharged state*); if a sensor node becomes a node with the discharged state, it does not change its state. Table 1 indicates the roles of a sensor node, according to its state; moreover, Figure 1 indicates the state transition diagram of EECSM.

4.2. Clustering Phase. The clustering phase commences when the sensor nodes are first scattered in the sensor field or after the completion of the "data transmission phase." This phase decides new CHs to form new clusters for the WSN. To achieve energy efficiency, the criterion in the selection of CH is the remaining energy of CMs. To reduce the load (or overhead) of CHs, the EECSM regulates the size of the clusters. The clustering phase is comprised of four steps: *broadcasting step*, *splitting step*, *CH selection step*, *and clustering step*, as follows.

4.2.1. Broadcasting Step. When the sensor nodes are scattered in the sensor field at period zero, there is no CH. Thus, all sensor nodes must decide on the CHs only on their own. The EECSM always uses the remaining energy, in order to decide the next CHs, except in period zero. Because the initial energies of all sensor nodes are identical in our assumption, all sensor nodes become CHs in period zero. To avoid this wasteful situation, EECSM uses another CH selection method just once at period zero. We define the *broadcasting range*, before the explanation of the other CH selection method. The broadcasting range is the reachable range of a data packet transmitted from a sensor node. The CH selection method at period zero uses the number of neighborhood sensor nodes within the broadcasting range, in order to decide who will be the CHs for the first round. In selecting a sensor node having the most neighbors within the broadcasting range as a CH, distances between the CH and CMs are reduced. This reduces the transmission energy between the CH and its CMs. Therefore, in order to decide CHs, we use the number of neighborhood sensor nodes within the broadcasting range just once at period zero. The procedure of deciding CHs at period zero is presented from 1.1 to 1.4.

To avoid selecting many CHs, the comparison range of the number of received signal packets widens the broadcasting range twice (from 1.3 to 1.4). The broadcasting step is not included in Figure 2, the flowchart of EECSM, since it is used only once at period zero.

4.2.2. Splitting Cluster Step. After each cluster configuration, EECSM considers the suitable number of clusters. First, EECSM tries to split a large cluster into two small clusters. The disadvantage of a large cluster is the high energy consumption of its CH, when the data packets are transmitted from its CM to its CH via a relatively long transmission path. When receiving the data packets, the more CMs the cluster has, the more its CH consumes energy. A processing bottleneck can also occur at the CH in a large cluster.

After the data transmission phase of the previous clustering round, in order to decide the number of clusters in the next clustering round, EECSM selects the next step, according to the value of the splitting threshold. If the number of CMs of a cluster is more than the splitting threshold value, the EECSM executes the splitting cluster step. Since a cluster is split into two clusters at the splitting cluster step, in the next round, two CHs for the split cluster are identified as the First CH and the Second CH, respectively.

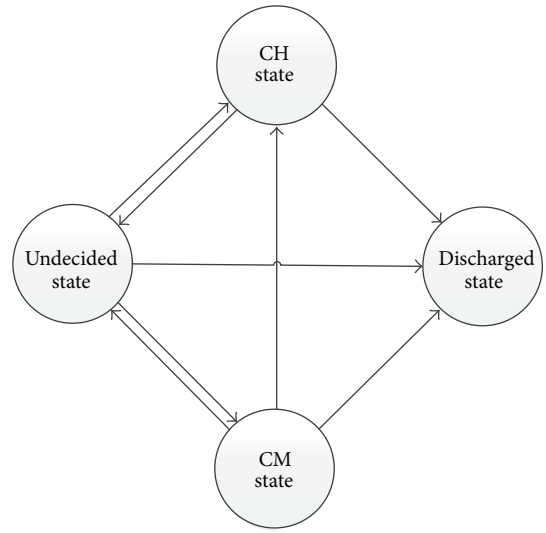


Figure 1: Transition diagram of sensor node states.

The procedure of deciding the First CH is described from 2.1 to 2.5 of Algorithm 1. The procedure of deciding the Second CH can be seen from 2.6 to 2.8 of Algorithm 1. The CH consumes energy much more than the CMs. If a sensor node having low energy becomes the CH, it can quickly reach the discharged state. To prevent this problem, EECSM selects a CM having maximum energy as the next round CH. To prevent locating the First CH and the Second CH in close proximity, the procedures from 2.3 to 2.5 of Algorithm 1 are executed.

4.2.3. CH Selection Step. If the number of CMs of a cluster is less than the splitting threshold value, the EECSM executes not the splitting cluster step, but the CH selection step. In this step, the CH decides only one CH for the next clustering round. The procedure of deciding the next round CH is presented from 3.1 to 3.2 of Algorithm 1. It is similar to deciding the First CH of the splitting cluster step.

4.2.4. Clustering Step. Each CH broadcasts a CH-signal packet to the entire sensor field for clustering. The sensor nodes, except for CHs, or nodes with discharged state receive, store, and list the CH-signal packets. Due to the fact that information can be used at the reclustering step of the merging cluster phase, the EECSM stores and lists the information temporally. This can reduce overhead of re-clustering. To minimize energy consumption of CMs during the data transmission phase, the sensor nodes select the nearest CH as its CH.

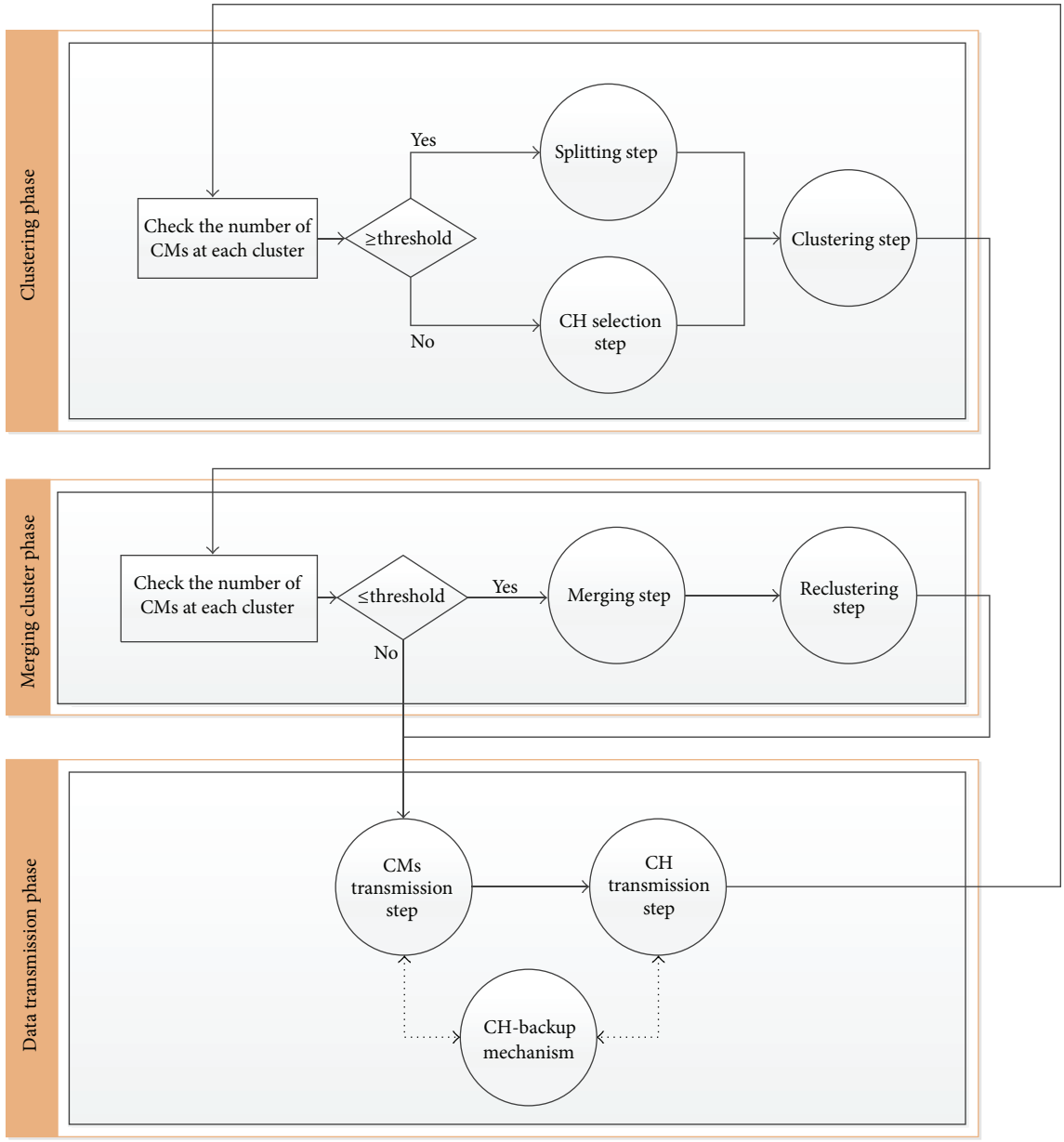


Figure 2: Flowchart of EECSM.

4.3. Merging Cluster Phase. After the clustering phase of the current clustering round, the EECSM checks the size of the clusters. Since the EECSM checks and splits the large clusters at the previous phase, it needs to check only the small clusters at this phase. The disadvantage of a small cluster is inefficient energy consumption, particularly when the data packets are transmitted from the CH to the BS, since energy consumption from a CH to the BS is much more than the energy consumption from a CM to the BS for longer transmission. For this reason, energy consumption of a small cluster is not energy efficient, and therefore the small clusters need to be merged into large clusters. The merging cluster phase on EECSM is comprised of two steps: *merging cluster step* and *re-clustering step*, as follows.

4.3.1. Merging Cluster Step. If the number of CMs of a cluster is less than the merging threshold value, the EECSM executes the merging cluster step. To merge small clusters into large clusters, procedures 1.1 to 1.2 are executed.

4.3.2. Reclustering Step. Procedure 2.1 is executed using the information stored in the clustering step of the clustering phase; after the re-clustering step is executed, the information stored in the clustering step is deleted (Algorithm 2).

4.4. Data Transmission Phase. If the merging cluster phase is finished, clustering is complete. The EECSM enters into the data transmission phase, as EECSM starts to inform

the situation of the sensor field to the external BS, by sending gathered data. The data transmission phase is divided into 2 steps: CM transmission step and CH transmission step.

In Algorithm 3, each CM creates a data packet that contains neighboring environment information for each period. The data packets are transmitted to the CH. Each CH aggregates the received data packets into a data packet and transmits the data packet to the BS directly. These procedures are repeated during each clustering round.

4.5. CH Backup Mechanism. WSNs are useful in inaccessible and dangerous areas, such as military targets, disaster regions, and hazardous environments: they protect humans from danger. Using sensor nodes in those areas may cause a breakdown of individual sensor nodes. This can reduce the network lifetime. Furthermore, the breakdown of CHs can affect the network lifetime, as well as entail a loss of information. If the CHs break or fail, information received from its CMs to the BS may be lost.

In order to reduce the degradation of the network lifetime, EECSM has a "CH backup mechanism." If the CH breaks or fails, the CMs that are close to their CH can recognize the breakdown of their CH. The reason why the CMs can realize the breakdown of their CH is that the CMs can recognize whether the data packet is transmitted from their CH to the BS. The advantage is that there is no additional overhead for recognizing the state of the CH.

The CH-backup mechanism of EECSM decides a new CH, after the breakdown of the CH. The CH backup mechanism is comprised of the following 2 steps: *CH reelection step* and *cluster recovery step*.

4.5.1. CH Reelection Step. The CH reelection step is carried out immediately, when the CM recognizes a breakdown of its CH during the data transmission phase (in procedure 1.1 of Algorithm 4). The CM recognizing a breakdown of its CH is usually the closest CM from the CH. In procedure 1.2 of Algorithm 4, the CMs broadcast the energy state-signal within the broadcasting range twice, in order to elect a new CH. The reason why that signal is broadcasted within the broadcasting range twice, rather than to the entire cluster, is to reduce the energy consumption of CMs. The CM having maximum residual energy becomes the new CH within that range.

4.5.2. Cluster Recovery Step. In procedure 2.1 of Algorithm 4, the new CH broadcasts the CH-signal to the entire area of the sensor field, since the new CH does not know the area of the cluster exactly. In procedure 2.3 of Algorithm 4, the CMs can decide their CH not only for the new CH of the cluster, but also for the CHs of other clusters, according to the distance. This can minimize the energy consumption of CMs during the data transmission phase.

The CH backup mechanism of EECSM can restore the unstable state of a network to the stable state of a network. It provides the robustness of the WSN.

5. Computational Experiments of EECSM

In this section, we show the performance of EECSM. To measure the performance of EECSM, we use Visual C++ of Visual Studio 2008.

5.1. Experimental Assumptions. In this section, we introduce the experimental environment of the EECSM.

(i) The locations of all sensor nodes and the BS are fixed.

(ii) The deployment of sensor nodes uses random distribution.

(iii) The location of the BS (x-axis: 25 m, y-axis: 150 m) is known in advance in the 50 m$*$50 m sensor field. For this field, the broadcasting range of EECSM is set to 10 m.

(iv) The data packet size is 1,000 bits, and the signal packet size is 50 bits.

(v) All sensor nodes have an initial energy of 0.5 J.

(vi) The test utilized the average of the performances of 10 different deployments of sensor nodes in the sensor fields.

(vii) We assumed a WSN cannot operate when more than 30% of the sensor nodes are discharged. Therefore, the network lifetime is defined as the time when 30% of sensor nodes are discharged in our experiments, except in the robustness experiment.

(viii) In the experiments, the number of sensor nodes is 100, except for scalability experiments.

5.2. Clustering with Splitting and Merging. The size of clusters and the number of clusters seriously affect energy efficiency and the network lifetime. To properly adjust the size of clusters and the number of clusters, EECSM merges small clusters into other clusters and splits each large cluster into two clusters. To prove the validity of clustering with the splitting and merging method of EECSM, we show the following experiments.

5.2.1. Setting of Splitting and Merging Threshold Value. In this experiment, we find the optimal values of the splitting threshold value and the merging threshold value. We have that the range of experiments of the splitting threshold value is 31 $\sim$ 60, and the range of experiment of the merging threshold value is 1 $\sim$ 30, for the cases of 100 sensor nodes. The result of the enumerative method is as follows: the optimal value of the splitting threshold value is 46, and the optimal value of the merging threshold value is 20. Therefore, from now on, EECSM splits a cluster that has more than 46 CMs into two clusters and merges a cluster that has less than 20 CMs into other clusters.

5.2.2. Relative Frequency of Splitting and Merging. To prove the validity of clustering with the splitting and merging method of EECSM, we show how many splitting cluster steps and merging cluster steps are used. In this experiment, we do

Clustering phase
(1) Broadcasting step
{**If (Period = 0)**}
(For all sensor nodes)
(1.1) Broadcast an undecided state-signal packet within the broadcasting range.
(1.2) Receive the undecided state-signal packets and count the number of received signal packets.
(For all sensor nodes)
(1.3) Broadcast the number of received signal packets within 2∗(broadcasting range).
(1.4) Receive the signal packets (from 1.3), and compare the number of received signal packets from others with the number of received signal packets of its own.
Within 2∗(broadcasting range), if one's value is Max, the sensor node becomes a CH.
Else
(For each cluster)
If (The number of CMs ≥ Splitting threshold)
(2) Splitting cluster step
(For each CH)
(2.1) The CH decides the First CH that is the CM having maximum residual energy of the cluster.
(2.2) The CH transmits a First CH-signal to the First CH.
(2.3) The First CH broadcasts a neighborhood-signal to neighbor sensor nodes.
(2.4) The CMs that receive a neighborhood-signal transmit ACK to the First CH.
(2.5) The First CH aggregates information of ACKs, and transmits to the CH. (List of neighbors)
(2.6) The CH decides the Second CH that is the CM having maximum residual energy of the cluster, except for the First CH and the CMs included in the list of neighbors.
(2.7) The CH transmits a Second CH-signal to the Second CH.
(2.8) The CH becomes a node with an undecided state.
Else
(3) CH selection step
(For each CH)
(3.1) The CH decides the next CH that is the CM having maximum residual energy of the cluster.
(3.2) The CH transmits a CH-signal to the next CH.
(4) Clustering step
(4.1) CH broadcasts a CH-signal packet to the entire area of the sensor field.
(4.2) Others decide their own CH, by finding the closest CH.

Algorithm 1: The clustering phase.

not count the number of splitting cluster steps or merging cluster steps per one clustering round, but rather count the number of clustering rounds when splitting cluster steps or merging cluster steps occur.

Table 2 indicates the average of relative frequency of splitting cluster steps and the average of relative frequency of merging cluster steps during network lifetimes. The splitting cluster step happens 60.3% of the total rounds, and the merging cluster step happens 55.0% of the total rounds. The results show that the splitting and merging of clusters happen frequently, in order to maintain the suitable number of clusters of EECSM, and this can seriously affect the network lifetime.

5.2.3. Comparison of Network Lifetime of Clustering with and without Splitting and Merging. To prove the validity of clustering with the splitting and merging method of EECSM, we show the network lifetimes of clustering with and without the splitting and merging method of EECSM.

Figure 3 demonstrates that the network lifetime of clustering with the splitting and merging method of EECSM is between 7.03% (when the 1st sensor node is discharged) and 11.21% (when the 30th sensor node is discharged) longer than clustering without the splitting and merging method of EECSM. This experiment indicates that clustering with the splitting and merging method of EECSM prolongs network lifetime, by maintaining the suitable size of clusters. The suitable sizes of clusters reduce energy consumptions of sensor nodes, by reducing the transmission range of sensor nodes.

5.3. Comparisons of Basic Performance with HEED. In this section, we compare the network lifetimes and residual energies of the sensor nodes of EECSM and HEED.

5.3.1. Network Lifetime Comparison. In this experiment, we compare the network lifetime of HEED and EECSM with the splitting and merging mechanism.

Figure 4 shows that the network lifetime of EECSM is 168.9% (when the 1st sensor node is discharged) longer and 23.7% (when the 30th sensor node is discharged) longer than the lifetime of HEED, respectively. This means that

Merging cluster phase
(For each cluster)
If (The number of CMs ≤ Merging threshold)
(1) Merging cluster step
(Each CH)
(1.1) Broadcast a merging cluster signal packet.
(1.2) All CMs and CH of the cluster become nodes with an undecided state.
(2) Re-clustering step
(2.1) All nodes with an undecided state decide on their own CH, by finding the closest CH.
Else
Do nothing

Algorithm 2: The merging cluster phase.

Data transmission phase
While (1 clustering round)
(1) CM transmission step
(Each CM)
(1.1) Sense environment.
(1.2) Make a data packet.
(1.3) Transmit a data packet to its CH.
(2) CH transmission step
(Each CH)
(2.1) Aggregate received data packets from its CMs
(2.2) Transmit a data packet to the BS.
End while

Algorithm 3: The data transmission phase.

the five considerations of EECSM using the energy information operate properly and also prolong the network lifetime.

5.3.2. Residual Energy Comparison. In this experiment, we compared the average ratios of the residual energies to the network lifetimes of EECSM and HEED.

In Figure 5, the average ratio of residual energy of EECSM is at most 13.0% of that of HEED. This experiment indicates that the sensor nodes of EECSM consume energy much more uniformly than do those of HEED.

Comparisons of the two experiments (Figures 4 and 5) indicate the following. First, the sensor nodes having maximum energy from the neighborhood are decided by the next CHs, while reducing the frequent selection of certain sensor nodes. This allows a variety of selection of CH and maintains fair balance in the energy of sensor nodes.

Second, proper splitting and merging of clusters reduce the transmission distance of sensor nodes. In these results, we conclude that EECSM reduces and maintains fair balance in energy consumptions of sensor nodes, with the consequence that it prolongs network lifetime.

5.4. Advanced Performance of EECSM. In this section, we show the scalability and robustness of EECSM.

5.4.1. Scalability. Because WSNs consist of hundreds or thousands of sensor nodes, clustering and routing protocols of WSNs should have scalability. Since EECSM is a self-organized clustering model, EECSM should have good scalability. In this experiment, the number of sensor nodes is increased, in order to prove the scalability of EECSM from 100 sensor nodes to 500 sensor nodes. Here, the network lifetime of the WSN is defined only as the time when 30% of the sensor nodes are discharged.

Figure 6 shows that the network lifetime of EECSM, when the number of sensor nodes is more than 200 sensor nodes, is longer than the lifetime of EECSM, when the number of sensor nodes is 100 sensor nodes, from a minimum 11.67% to a maximum 14.68%. According to the increase of the number of sensor nodes, EECSM tends to increase the network lifetime, rather than to degrade the network lifetime. We can conclude that EECSM has good scalability, since the splitting and merging method of EECSM reduces the transmission range and maintains fair balance in energy consumptions of sensor nodes.

5.4.2. Robustness. WSNs are useful in dangerous areas, such as military targets, disaster regions, and hazardous areas. Thus, sensor nodes of WSNs in these areas can break suddenly. To use WSNs in these dangerous areas, clustering and

CH backup mechanism
(For each cluster)
If (CH is dead; breakdown or discharged of energy)
(1) CH reelection step
(1.1) A CM that detects a failure of its CH broadcasts a CH-failure-signal
(For each CM received the signal)
(1.2) The CMs broadcast an energy state-signal within
2∗broadcasting ranges
(1.3) The CMs compare the energy states
The CM having maximum energy becomes the new CH
(2) Cluster recovery step
(2.1) The CH broadcasts a CH-signal packet to the entire area of the sensor field
(For each CM of the cluster)
(2.2) Receive the CH-signal packet
(2.3) The CMs compare the distances
If (the distance between the CH of the cluster and the CM ≤ The distance between the closest CH of other cluster and the CM)
The CM becomes a CM of the cluster
Else
The CM becomes a CM of the other closest cluster

Algorithm 4: The CH-backup mechanism.

Table 1: State of sensor nodes.

State of sensor node		Specification
Undecided state	Condition	When a sensor node has been scattered first in the sensor field or a sensor node has acted as a CH or CM, the sensor node becomes an undecided state node.
	Act	An undecided state node awaits signal packets from the CHs, except when a sensor node has been scattered in the sensor field for the first time.
CH state	Condition	(1) When the number of CMs of a cluster is less than the splitting threshold, one CM becomes the CH state node. (2) When the number of CMs of a cluster is more than the splitting threshold, two CMs become CH state nodes. (3) When a CH breaks or fails, a node with the CM state within the broadcasting range of the CH becomes a CH state node.
	Act	A CH collects and aggregates information from its CMs and then transfers data packets to the BS. Also, a CH decides the CH(s) for the next clustering round.
CM state	Condition	When a CH is decided, all nodes with the undecided state in the cluster of the CH become CM state nodes.
	Act	A CM periodically sends sensed information to its CH.
Discharged state	Condition	When a sensor node cannot function anymore, because all of its energy has been discharged, or it has broken down, it becomes a discharged state node.
	Act	Nothing is done.

routing protocols should be able to restore the network when a break happens. In order to prove that the CH backup mechanism works properly, we conduct the following experiments.

In this experiment, we compare the network lifetimes of normal states and abnormal states. The normal state means that a WSN is operating normally, without sudden breakdowns of sensor nodes. The abnormal state means that 0, 10, 20, 30, or 40 sensor nodes suddenly broke, sporadically, in the WSNs, before the 2,000th period.

In this experiment, the network lifetime is defined as the time when 50% of the sensor nodes which malfunctioned. 50% of sensor nodes malfunctioned is the total percentage of 100 sensor nodes that have encountered sudden breakdowns or discharged their energy; for example, when the number of

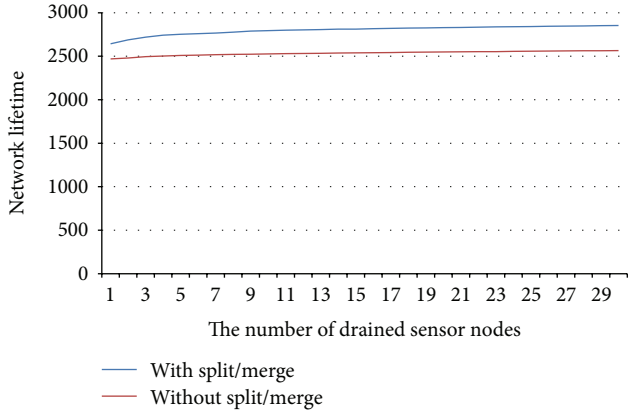


Figure 3: Comparison of network lifetimes of clustering with and without the splitting and merging method of EECSM.

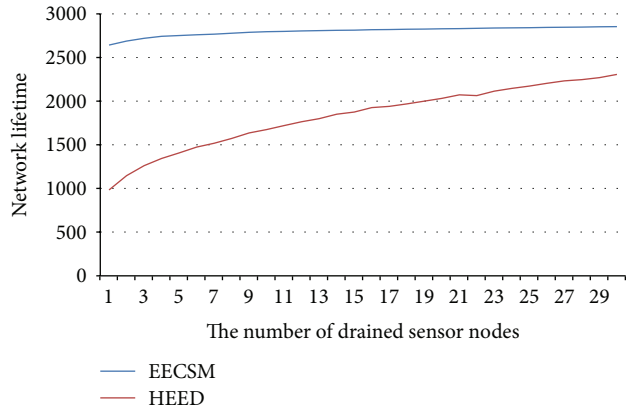


Figure 4: Comparison of network lifetimes of EECSM and HEED.

Table 2: The relative frequency of merging and splitting.

	Splitting cluster step	Merging cluster step
The average of steps	173.3	158.1
The average of clustering rounds	287.2	287.2
The average of relative frequency	60.3%	55.0%

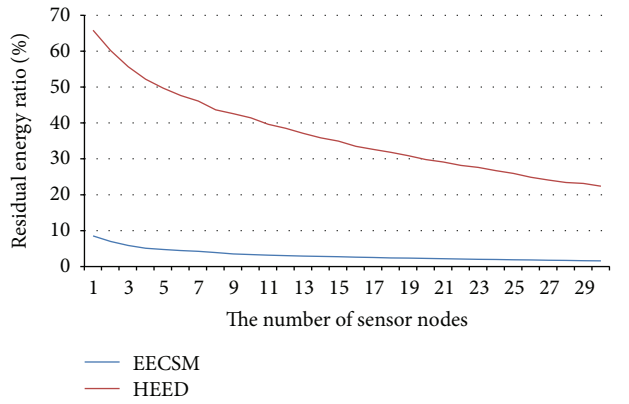


Figure 5: Experiment on the residual energy ratios of EECSM and HEED.

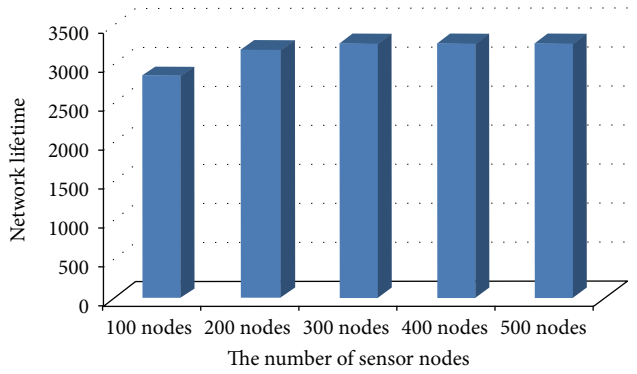


Figure 6: Experiment on the scalability of EECSM.

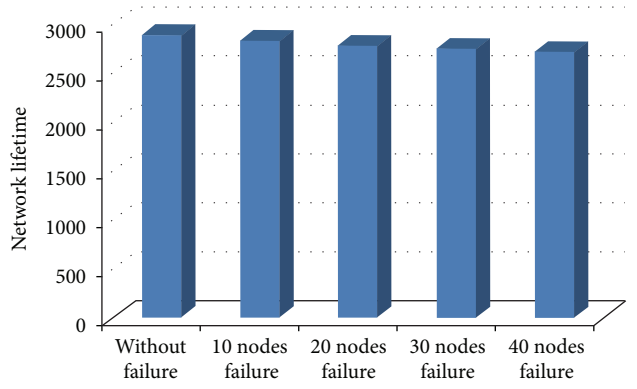


Figure 7: Experiment on sporadic failures.

sensor nodes are 100, 50% of sensor nodes which malfunctioned consist of sensor nodes discharged of their energy (A) and sensor nodes which suddenly broke down (B). (A, B) is (50, 0), (40, 10), (30, 20), (20, 30), or (10, 40).

Figure 7 shows that, although unexpected breakdowns occur, the effect is not serious. The maximum degraded network lifetime, when 40 sensor nodes suddenly break down, is at most 6.30%. The CH backup mechanism can restore the unstable state of the network to the stable state of a network, without any additional overhead. It can minimize the energy consumption of the unstable state and the CH backup mechanism, also. Therefore, the CH backup mechanism of EECSM is a robust mechanism, capable of coping with abnormal conditions of the external environment.

6. Conclusion

This paper has proposed an energy-efficient self-organized clustering model, the so-called EECSM. The proposed model attempts to maximize the network lifetime and maintain load

balance through the selection of CHs and by resizing clusters through combined techniques of advantages of self-organized protocols and cluster-based routing. Since EECSM uses a self-organizing approach, it has good characteristics, such as distributed control, adaptability, robustness, and scalability. Moreover, EECSM can decide on the proper CHs for energy efficiency. It can also resize clusters for maintaining a suitable size, and further it can also restore damaged clusters on its own, based on local information.

Clustering with the splitting and merging method has many interesting research issues. Appropriate research examples are the self-decision method of the merging threshold and the splitting threshold through information, which are localized neighbor information, for example, the number of neighbors, the current state of neighbors, and so on. In addition, research on clustering with the splitting and merging method for mobile ad-hoc networks (MANETs) holds the promise of valuable results.

Acknowledgments

This paper is the result of the research with the support of National Research Foundation of Korea (2009-0074081), using the funds from the Ministry of Education, Science and Technology, in 2009.

References

[1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[2] S. Mahfoudh and P. Minet, "Survey of energy efficient strategies in wireless ad hoc and sensor networks," in *Proceedings of the 7th International Conference on Networking (ICN '08)*, pp. 1–7, April 2008.

[3] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: a survey," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6–28, 2004.

[4] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, vol. 3, no. 3, pp. 325–349, 2005.

[5] N. Israr and I. U. Awan, "Multilayer cluster based energy efficient routing protocol for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 4, no. 2, pp. 176–193, 2008.

[6] M. Younis, M. Youssef, and K. Arisha, "Energy-aware management for cluster-based sensor networks," *Computer Networks*, vol. 43, no. 5, pp. 649–668, 2003.

[7] F. Dressler, *Self-Organization in Sensor and Actor Networks*, John Wiley & Sons, New York, NY, USA, 2007.

[8] C. Prehofer and C. Bettstetter, "Self-organization in communication networks: principles and design paradigms," *IEEE Communications Magazine*, vol. 43, no. 7, pp. 78–85, 2005.

[9] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Annual Hawaii International Conference on System Siences (HICSS-33 '00)*, January 2000.

[10] Y. Sun and X. Gu, "Clustering routing based maximizing lifetime for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 5, no. 1, pp. 88–88, 2009.

[11] O. Younis and S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, 2004.

[12] O. Younis, S. Fahmy, and P. Santi, "An architecture for robust sensor network communications," *International Journal of Distributed Sensor Networks*, vol. 1, no. 3-4, pp. 305–327, 2005.

[13] X. Ji, H. Zha, J. J. Metzner, and G. Kesidis, "Dynamic cluster structure for object detection and tracking in wireless ad-hoc sensor networks," in *Proceedings of IEEE International Conference on Communications*, vol. 7, pp. 3807–3811, June 2004.

[14] Y. C. Wang, W. C. Peng, and Y. C. Tseng, "Energy-balanced dispatch of mobile sensors in a hybrid wireless sensor network," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 12, pp. 1836–1850, 2010.

[15] H. Jiang, S. Jin, and C. Wang, "Prediction or not? An energy-efficient framework for clustering-based data collection in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 6, pp. 1064–1071, 2011.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 205192, 9 pages
http://dx.doi.org/10.1155/2013/205192

Research Article

Traffic Priority and Load Adaptive MAC Protocol for QoS Provisioning in Body Sensor Networks

Iffat Anjum,[1] Nazia Alam,[1] Md. Abdur Razzaque,[1] Mohammad Mehedi Hassan,[2] and Atif Alamri[2]

[1] *Department of Computer Science and Engineering, Faculty of Engineering and Technology, University of Dhaka, Dhaka-1000, Bangladesh*
[2] *Information Systems Department and Chair of Pervasive and Mobile Computing, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia*

Correspondence should be addressed to Md. Abdur Razzaque; razzaque@cse.univdhaka.edu

Received 26 October 2012; Revised 10 February 2013; Accepted 17 February 2013

Academic Editor: Sheikh I. Ahamed

Body sensor networks (BSNs) carry heterogeneous traffic types having diverse QoS requirements, such as delay, reliability and throughput. In this paper, we design a priority-based traffic load adaptive medium access control (MAC) protocol for BSNs, namely, PLA-MAC, which addresses the aforementioned requirements and maintains efficiency in power consumption. In PLA-MAC, we classify sensed data packets according to their QoS requirements and accordingly calculate their priorities. The transmission schedules of the packets are determined based on their priorities. Also, the superframe structure of the proposed protocol varies depending on the amount of traffic load and thereby ensures minimal power consumption. Our performance evaluation shows that the PLA-MAC achieves significant improvements over the state-of-the-art protocols.

1. Introduction

Wireless body sensor networks (BSNs) are getting immense research interests worldwide as they make a paradigm shift in medical applications, and, in particular, health monitoring systems. With the advent of miniature, cost effective, and wearable sensor devices, they have attracted large amount of research time [1]. BSNs provide the scope of early detection of critical health conditions and diseases. Nowadays, people suffer from many chronic diseases that require constant monitoring of health condition. By making the use of wireless body sensor network, a patient can be under constant monitoring without any restriction and expenses of being in a hospital. This process can be considered as the next step in enhancing the personal health care and in coping with the costs of the health care systems [2]. Also, they can be deployed inexpensively in existing structures without IT infrastructure [3]. The BSN technology can be used to help protect those exposed to potentially life-threatening environments, such as soldiers, first responders, and deep-sea and space explorers [4]. It is also being used successfully for entertainment applications [5].

A BSN comprises of a number of biomedical sensor nodes which are placed on or implanted in a human body. These sensors collect various physiological data and transmit them to a coordinator node which in turn transmits the data to the main server via the Internet. BSNs primarily use a star topology (Figure 1) with a communication range of around 3 meters [6], and the sensors usually need to transmit data at relatively wide range of data rates from 1 Kbps to 1 Mbps [7]. The two fundamental design challenges in BSNs are energy efficiency and quality of service (QoS) [8]. And BSNs, which are deployed to permanently monitor human physiological parameters, must satisfy far more stringent quality of service (QoS) demands than those of other existing wireless sensor networks [9]. Provisioning QoS such as reliability and timely delivery is very crucial for BSN. Most of the time, these sensors are very small in size and have low battery power; so, energy efficiency should also be maintained. Priority awareness is a significant matter in BSN as the various

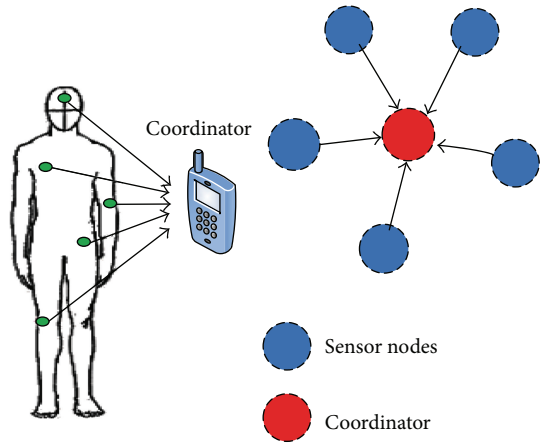


Figure 1: Network topology.

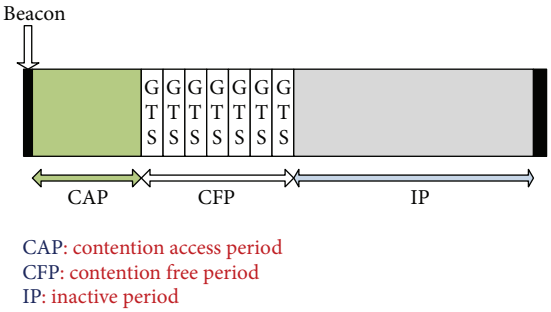


Figure 2: Superframe structure of 802.15.4.

data collected by the sensors can have different degree of importance and the resources are limited; so, emergency and highly important data packets should get a better chance to be transmitted.

A number of medium access control (MAC) protocols have been proposed for provisioning QoS in BSN. IEEE 802.15.4 [10–12] (Figure 2) is a standard defining the specifications for the MAC layer of a low rate wireless personal area network (WPAN), which also provides a way for QoS provisioning in BSN. But the superframe structure of IEE 802.15.4 is not flexible and also the latency involved is high. BodyQoS [13] implements a virtual MAC to schedule and represent channel resources, which makes it radio-agnostic. But there is no priority consideration and also high computational complexity is involved. ATLAS [14] proposes a traffic load aware MAC protocol where the structure of the superframe depends on the estimated traffic load. But it does not take priority into account. PNP-MAC [15] adopts the superframe structure of IEEE 802.15.4. It proposes an MAC protocol with preemptive slot allocation and nonpreemptive transmission and also takes priority into account. But it does not take traffic load into account; duration of the CFP period of its superframe structure is fixed.

In this paper, we propose a priority-based load aware MAC protocol for BSN where the data packets are served based on their priority and the superframe structure is dynamic. In the proposed MAC protocol, we categorize the data packets into four priority classes based on their data type and generation rate. Then, these packets are served according to their priority values. So, high priority and emergency packets are served first. The superframe structure of the protocol varies based on the traffic load as the number of slot allocations in the superframe structure depends on the traffic load. So, the superframe structure contains a large inactive period when the traffic load is low and the opposite when the traffic load is high. So, energy efficiency is also ensured.

The rest of the paper is organized as follows. In Section 2, we have discussed some related works and their limitations. Then, we present the network model in Section 3. In Section 4, we describe in detail the proposed priority-based and traffic load adaptive MAC protocol for body sensor networks. The performance evaluation and results are presented in Section 5. Then, we conclude the paper in Section 6.

2. Related Works

The IEEE 802.15.4 [10–12] standard was specifically devised to support low power, low data rate networks where latency and bit rate are not so critical. Although the design environment was different for IEEE 802.15.4, as a response to the growth in wireless personal area network, many earlier works [16] adopted the IEEE 802.15.4 MAC protocol and its superframe structure to support the QoS requirements of BSNs. The superframe structure of 802.15.4 consists of a contention access period (CAP), a contention free period (CFP), and an inactive period (IP). The CFP period contains up to seven guaranteed time slots (GTSs), which limits the adaptive operation of BSN applications. Secondly, IEEE 802.15.4 does not have any mechanism for prioritizing among different applications, and low priority data can block the transmission of the high priority one, which can cause a severe problem in BSN. Thirdly, the requested GTS time slots are not allocated in the current superframe; they are scheduled to the next superframe, which increases the packet delay or latency.

LDTA-MAC [16] protocol improves some of the shortcomings of IEEE 802.15.4. The guaranteed time slots (GTSs) are not fixed, allocated dynamically based on traffic load. And also on successful GTS request, data packets are transmitted in the current superframe. But there is no consideration of the priority of the applications or backoff value of them. Also the CFP's and IP's durations are fixed.

BodyQoS [13] separates QoS scheduler from the underlying MAC implementation, and thus it does not suffer from the limited number of GTSs. However, BodyQoS uses nonpreemptive slot allocation schemes; so, high priority applications can be blocked by low priority applications. Also, separate MAC implementation can increase computational complexity.

ATLAS [14] proposes a traffic load aware MAC protocol where the superframe structure varies based on the estimated traffic load and uses a multihop communication pattern. But it does not take the priority of different applications into account. There is also no indication of backoff class depending on the priority to avoid collision and to let higher

Table 1: Data type.

Traffic-class value	Traffic class
4	Ordinary data packets (OPs)
3	Delay-driven data packets (DPs)
2	Reliability-driven data packets (RPs)
1	Critical data packets (CPs)

priority application request first. Also, managing four types of adaptive superframe structure depending on traffic load may become a computational load on the gateway.

PNP-MAC [15] protocol is based on IEEE 802.15.4 superframe structure. It can flexibly handle applications with diverse requirements through fast, preemptive slot allocation, nonpreemptive transmission, and superframe adjustments. But the duration of CFP is fixed; so, if we have lower data rate, it will cause loss of time and energy for being awake on this period. Again if the data rate is high, then the fixed inactive period (IP) can cause many important and high priority data to wait for the next superframe. Here, if any data is generated during the IP, then it will be lost or have to wait, which is not viable for emergency data. Again the CAP period only takes request for GTS, not data packets; for some application, this period of delay can be significant. There is no balance between the priority consideration and traffic load of sensor nodes. As low priority sensors can have a higher traffic load and they have greater backoff, they will not be able to send most of the data and drop them, which can cause a major problem in medical treatments.

3. Network Model and Assumption

In the proposed PLA-MAC protocol, we assume that several biomedical sensor devices are attached to a human body; they all collect data and transmit the data to a central coordinator node using a star topology. The coordinator node can be a Smartphone, a Smart watch, or PDA, which will transmit the data to the external network. The sensor nodes are assumed to have limited energy supply and limited processing power. The coordinator is significantly more powerful than the sensor nodes. Therefore, it is desirable to push as much computation and communication overhead to the coordinator as possible.

In addition, considering the normal application scenario of a BSN such as a data collection system where data are sent from the sensor nodes to coordinator, the down link traffic like notification or beacon from the coordinator is not considered significant.

We assume that every data packet has a lifetime T_{life}, specified by an application, which indicates the time limit within which the packet should be delivered to the coordinator; otherwise, the information in the packet is useless, and it should be dropped.

In the proposed PLA-MAC protocol, the superframe structure is a modified version of the superframe of the IEEE 802.15.4 protocol.

A proper description of the superframe structure used in PLA-MAC can be found in Section 4.3. Here, we have assumed the superframe to have a dynamic structure; the length of the active part of the superframe changes based on the traffic load in the network. The superframe is assumed to contain a fixed CAP of 20 slots, and the length of the superframe is 128 slots. The number of CFP slots is not fixed. So, when there is minimal traffic, the length of the active part of the superframe can be just a little more than 20, and when there is a huge traffic load, the length of the active part of the superframe can be near 128. The rest of the superframe will inactive, which is specified by IP (inactive period).

4. Proposed MAC Protocol

4.1. Overview. In this paper, we propose a priority-based MAC protocol for body sensor networks that modifies the superframe structure of IEEE 802.15.4. It has a dynamic superframe structure depending on the variation of traffic loads. Based on the delay and reliability constraints of data packets, we primarily perform a traffic classification. Using this classification and data generation rates from sensor nodes, we calculate the different *priority* and *backoff* values. The priority class is used by the coordinator while allocating slots for data packets. The backoff values are used by the sensor nodes to perform prioritized random backoff before transmitting the data packets.

4.2. Traffic Classification. In this protocol, the generated data packets are divided into four types: ordinary data packets (OPs), delay-driven data packets (DPs), reliability-driven data packets (RPs), and critical data packets (CPs) [17] (Table 1). The OP corresponds to a data packet that contains regular physiological measurements like body temperature, which does not have any strict reliability or delay constraints. The DP corresponds with packets that have to be delivered timely but do not have much reliability constraint, for example, video streaming. The RP packets must be delivered with high reliability that is without any loss of data, but do not have any delay deadline, for example, respiration monitoring and PH monitoring. The CP packets have high reliability and delay constraints; they have to be delivered with higher reliability and lower end-to-end delay, for example, ECG data packets. Note also that the CP packets have the highest data generation rate and packet size compared to other classes; OP packets have lowest data generation rate and packet size compared to other classes of packets.

Here, the data packets are assigned a data type number T_i. The critical data packets are assigned the lowest and ordinary packets are assigned the highest data type number. Based on this data type number, the corresponding backoff and priority values will be calculated.

4.3. Superframe Structure. The BSN superframe structure contains five periods: beacon, contention access period (CAP), notification, contention free period (CFP), and inactive period.

In Figure 3, we can see that every superframe starts with a beacon period. The beacon informs all member nodes about the basic information about the coordinator, other nodes,

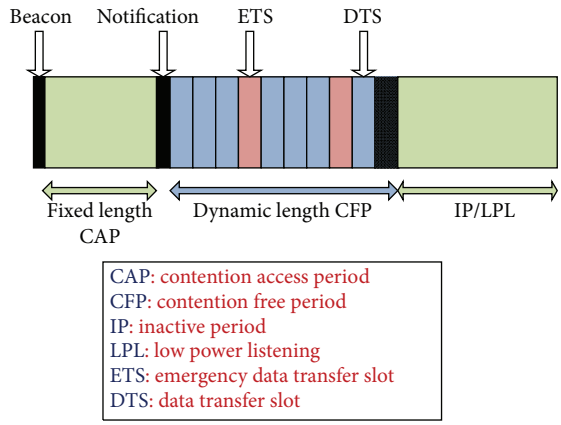


Figure 3: BSN superframe structure.

and about the start of the superframe. In the CAP period, the allocation requests for the CP, RP, and OP classes of data packets for the CFP slots, and data packets of the DP classes are received by the coordinator from sensor nodes. Whether a node will transmit a DP packet or a DP request in CAP period is totally implementation dependent. After CAP, the coordinator allocates DTS slots to the packets, and the allocation status is announced through a notification to all the sensor nodes. The coordinator node allocates DTS slots to the packets based on their priority values. Higher priority packets are allocated first. Then, there is a CFP period, during which the allocated packets are transmitted. Here, we have a dynamic size of CFP; so, the CFP can be very small when there is very low traffic load or it can occupy the rest of the superframe when there is a high traffic load.

If the CFP does not occupy the entire superframe, then rest of the period is inactive period. This inactive period can be optionally used as low power listening (LPL).

In the CFP, a number of ETS slots are kept for transmitting emergency packets that are generated after the CAP period. Nodes having emergency packets perform clear channel assessment (CCA) to occupy an ETS slot.

If a sensor node cannot send slot allocation request to the coordinator successfully during the CAP period, the transmission of packets from those nodes will be handled as follows.

(i) As described earlier, the emergency packets will be sent in the ETS time slots, after doing a CCA.

(ii) For RP and OP packet types, they will be stored in the buffer of the corresponding node, waiting for slot allocation in the next superframe. Such packets may be dropped if the node buffer overflows or packet lifetime (T_{life}) exceeds.

4.4. Backoff Calculation. Prioritized random backoff is performed in CAP. A node, which sends either a data packet or a request packet, performs a random backoff. The backoff value is chosen from the range $[0, 2^{T_i+2} - 1]$, where T_i is the traffic class number. So, the probability of a critical data packet to enjoy less delay is higher than other data packets.

As the backoff value is calculated based on the traffic class value, data packet that has lower traffic class value will get small backoff value and have to wait small period of time before sending a data packet or request. And data packets with higher traffic class value will get larger backoff value. For example, assume that we have a critical data packet and an ordinary data packet to send. As traffic class value for CP is 1, it will get a random backoff value between the range of [0, 7]. The traffic class value for OP is 4; so, its backoff will be in the range of [0, 63]. As in most cases, OP will have higher backoff value than CP; so, request of CP will be sent before OP.

4.5. Priority Calculation. The sensor nodes calculate the priority of each packet using the following equation:

$$P_i = \frac{T_i}{G_i \times S_i}, \tag{1}$$

where P_i = priority, T_i = traffic class value, G_i = data generation rate, and S_i = size in bytes.

We can see that the priority value of a packet depends on its traffic class value and the data generation rate of the node. Based on the priority calculated earlier, the packets will be categorized to be in one of the four following classes:

(i) emergency;

(ii) high;

(iii) average and;

(iv) low.

The packets with the lowest traffic class value (critical packets) and highest data generation rate will have the lowest score and highest priority, and they will be defined to be in emergency class. The significance of doing this is that the packets with low traffic class value contains, the most important data which must be delivered timely and with reliability, and packets that are from a node with high data generation rate also must be delivered quickly as the buffer of the sensor node will overflow otherwise and data packets will be lost. Similarly, the packets with the highest traffic class value and lowest data generation rate will have the highest score and lowest priority, and they will be defined to be in low class. The data packets with priority values in between these two classes will be defined to be in high and average classes depending on their values. The range of the priority classes is application dependent.

The coordinator node will allocate slots in the CFP period for the sensor nodes based on aforementioned priority, classes. Emergency packets are given the highest priority and they are allocated slots before any other packets. When slot allocation for emergency packets is finished, then slots for high priority packets are allocated. Average priority packets are allocated next and followed by low priority packets. The rest of the superframe structure is considered as inactive period.

4.6. Protocol Operation. As sensor nodes have low battery life, computational load should be kept as small as possible.

Also, each sensor node has a small buffer. If it has a data packet to send, the sensor stores it in the buffer and waits in low power listening mode for next beacon signal from coordinator. Each data packet has a maximum lifetime T_{life} within which it has to reach the coordinator; otherwise, the sensor node drops the packet.

The BSN superframe structure has been designed to be flexible depending on the traffic need. Every superframe starts with a beacon signal. After the beacon signal, the contention access period (CAP) starts, in which the nodes having CP, RP, and OP type data packets make requests for DTS slots. CP and RP data packets are sent using reserved time slots since they are loss-sensitive ones. On the other hand, the loss-insensitive DP type data packets content with each other to be transmitted in the CAP slot. In CAP, the receiver node sends back an acknowledgement (ACK) message after a packet is successfully received. This strategy is much helpful for DP packets to reach the coordinator node within their lifetime.

The sensor nodes may also request DTS slots for sending DP during the contention free period (CFP), depending on the applications need. The requests for DTS slots, which have been received in CAP, are first sorted by the coordinator node based on their priority values and then allocated accordingly. The coordinator node sends this allocation information to all nodes in the notification period, and thus the sensors get to know whether their requests have been granted or not; it also informs the slot number if a request is granted. Therefore, there is no need for sending ACK for every request; the notification does that part. A sensor node can *sleep* in the CAP period if it has nothing to send. The other nodes, after sending data or request packet, will go to *LPL* and wait for receiving any ACK (if data packet is sent) or notification (if request for DTS is sent). These *sleep* and *LPL* periods save the energy of sensor nodes.

The coordinator node allocates the DTS slots based on the priority of the requests. After completion of the transmissions in DTS slots during the contention free period (CFP), the BSN superframe proceeds with inactive period (IP) or LPL period. In IP, the coordinator node goes to sleep mode, and the sensor nodes turn off their transceiver circuitries, saving energy. In LPL, the sensor nodes might transmit emergency data packets only, depending on the implementation. Whether the IP or LPL will be activated can be determined dynamically by the coordinator node based on the traffic load of the network. Also, the IP may not be present in the superframe structure at very high traffic load conditions.

We also keep provisions of transferring emergency packets during contention free period by allocating few emergency time slots (ETSs). More specifically, the ETSs are for transmitting emergency data packets that are generated after the CAP period. Here, the number of ETSs can be calculated using exponential weighted moving average in the following way:

$$\text{NumETS} = (1 - \alpha) \times \text{NumETS} + \alpha \times \text{NumEMR}. \quad (2)$$

Here, the value of NumETS is a weighted combination of the previous value of NumETS and the last value of NumEMR, which is the number of emergency data packets received in the last superframe. In this way, the number of ETSs will be dynamically adjusted during each superframe according to the number of emergency data packets received in the most recent superframe. So, the number of ETSs will increase when a large number of emergency data packets are generated and decrease when the number of emergency data packets goes down.

Table 2: Simulation parameters.

Parameter	Value
Channel data rate	250 kbps
Payload size	32 bytes
MAC header	8 bytes
PHY header	6 bytes
Superframe period	1 s
Number of slots in a superframe	128
Slot duration	7.68 ms
Beacon order	6
CAP duration in IEEE 802.15.4 and PNP-MAC	8 slots
CAP duration in proposed protocol	20 slots
CFP duration of PNP-MAC	40 slots
Simulation time	100 s

5. Performance Evaluation

In this section, we compare the performance of the proposed priority aware and traffic load adaptive MAC protocol with PNP-MAC [15] and IEEE 802.15.4 [10]. Here, we have conducted simulation using a simulator which we have been implemented using object-oriented programming language C++. To analyze the performance of the studied protocols, we have compared them in the fields of average packet delay, throughput, and energy consumption.

5.1. Simulation Model. For the simulation of the aforementioned MAC protocols, we consider a body area sensor network consisting of a single coordinator and a number of sensor devices. The sensor devices collect data and transmit them to the coordinator using a single-hop star topology. The superframe parameters used in this simulation are BO = 6, slot size = 7.68 ms, number of slots = 128, and the CAP size of proposed MAC protocol = 20. We consider the CFP size of the PNP-MAC [15] protocol as 40 slots in the simulation as there was no specific size mentioned in the paper. For the proposed MAC protocol, we consider the number of DP packets to be 20% to 30% of the total data packets and the CP, RP, and OP packets to be the other 70% to 80%.

The network parameters used here are summarized in Table 2. The parameters used for evaluating power consumption are given in Table 3.

5.2. Performance Metrics. The following four metrics have been considered for the performance evaluation of our proposed PLA-MAC.

Table 3: Power consumption parameters.

Operation mode	Power consumption
Transmit	10 mA
Receive	4 mA
Sleep	20 uA
LPL	1 mA

(i) *Average Packet Delivery Delay.* In our network, there are several sensor nodes and a coordinator. Each of the nodes transmits packets, which are received by the coordinator. Packet delivery delay here is the time between generation of a packet at a sensor node and its reception at the coordinator node in specific slot of the corresponding superframe.

(ii) *Average Delivery Delay for Delay Driven Packets.* Delay-driven packets correspond to packets that have to be delivered timely. If they are not delivered in time, they are useless. In our protocol, we have given special treatment to the delay-driven packets; the delay-driven packets are transferred in the CAP period; so, they do not have to make request and then transmit in CFP slots.

(iii) *Throughput.* In communication networks, throughput or network throughput is the average rate of successful packet delivery over a communication channel. This data may be delivered over a physical or logical link or pass through a certain network node. The throughput is usually measured in bits per second (bps) and sometimes in data packets per second or data packets per time slot. In our evaluation, we have used kbps (kilobits per second); we have calculated the amount of payload bits carried in the total number of data packets received at the coordinator node.

(iv) *Coordinator Power Consumption.* Coordinator power consumption is the amount of power consumed at its different states, like transmit, receive, sleep, and low power listening states. As the architecture and orientation of the superframe in PLA-MAC is different than IEEE 802.15.4 and PNP-MAC, there is a significant difference in amount of power consumption as well.

5.3. Simulation Results. In our simulation performance evaluation, we study the impacts of the number of end devices and the amount of traffic loads from different devices.

5.3.1. Impacts of the Number of End Devices. Here, we vary the number of end devices from 1 to 10. Each node generates data at the rate of 5 Pkts/s.

First, we measure the average packet delay, the time needed to transmit a data packet to the coordinator shown in Figure 4. We can see that the average delay increases with the increase in the number of nodes; the reason behind that is the increased traffic and collision. In IEEE 802.15.4, as the GTS allocation information for the requests received in the current superframe is broadcasted in the beacon of the next superframe, the sensor has to wait for the next superframe to transmit data. So, the delay for IEEE 802.15.4 protocol is quite long. In the PNP-MAC protocol, the sensor nodes can transmit data in the same superframe in which they make request for slots; so, the delay is much less than the IEEE 802.15.4. But as PNP-MAC contains a fixed number of GTS slots, the delay increases when the number of data packet exceeds the number of GTS slots. However, in PLA-MAC, the CFP duration is not fixed and depends on the traffic load, and also the DP packets are transmitted in the CAP period; thus,

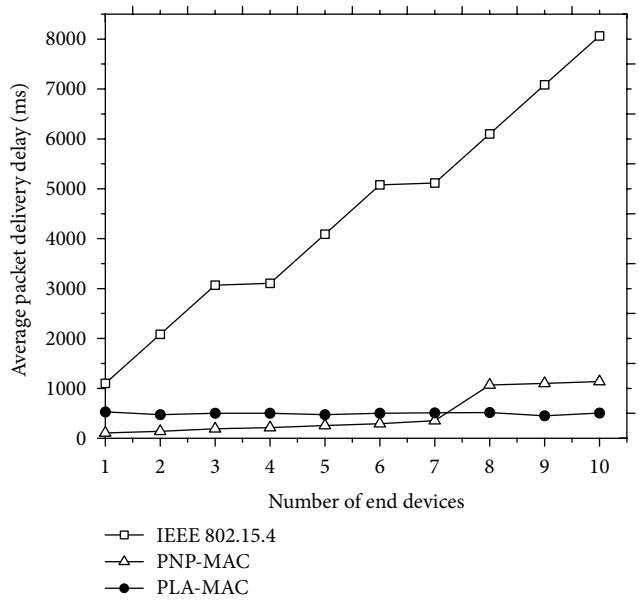


Figure 4: Average packet delivery delay versus number of end devices.

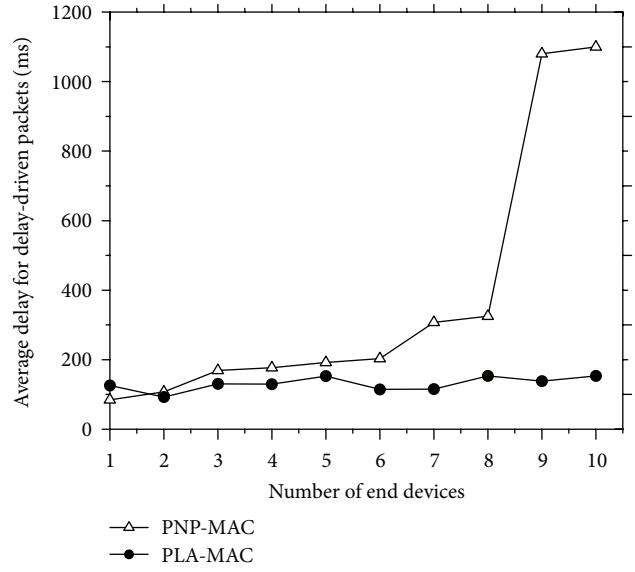


Figure 5: Average packet delivery delay for delay-driven packets versus number of end devices.

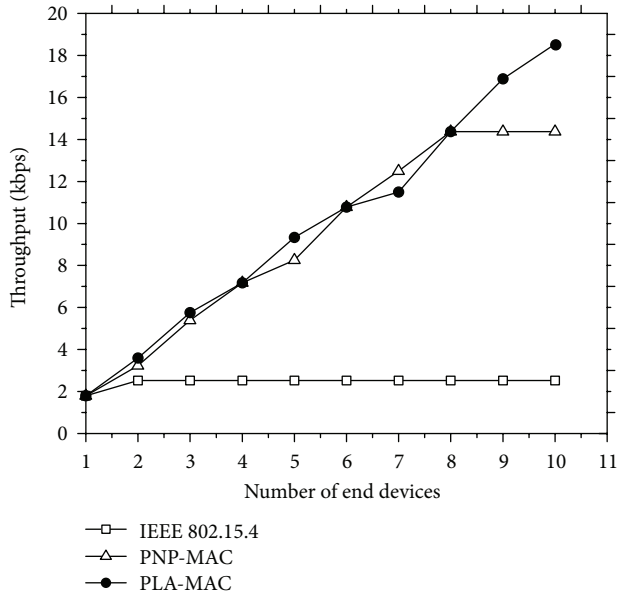


Figure 6: Throughput versus number of end devices.

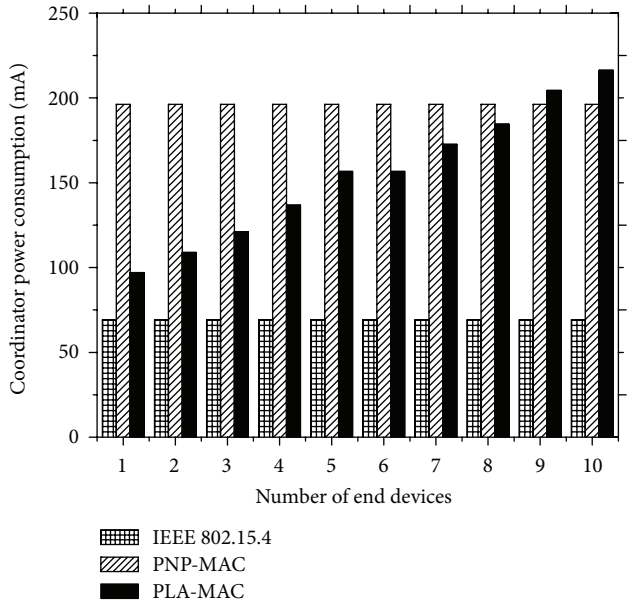


Figure 7: Coordinator power consumption versus number of end devices.

the waiting time of a packet is reduced, and the average packet delivery delay is minimized even when there is a large amount of data traffic in the network.

In Figure 5, we plot the average delay for the delay-driven packets. These packets have delay constraints, and they need to be delivered to the coordinator node within their lifetime. The IEEE 802.15.4 does not contain any special mechanism for handling delay-driven packets; so, they are transmitted in the same way the other packets do. The PNP-MAC protocol has a priority classification, and the packets are transmitted based on their priority values; still the sensor nodes have to send requests first and then the packets are transmitted in the allocated GTS slots. Furthermore, in case an allocation fails, the packet has to wait for the next superframe, and thus the delay increases a lot. However, in PLA-MAC, sensor nodes do not have to send requests and wait for slot allocation for the DP packets; rather they can send the DP packets in the CAP period, minimizing the delay considerably.

Next, we evaluate the throughput of the studied protocols in Figure 6. Throughput is the amount of data packets received by the coordinator in a specific time unit. Here, the throughput of all protocols increases with the increase in number of nodes. We can see that, as the IEEE 802.15.4 has only 7 GTS slots, the throughput becomes constant when the limit of 7 slots is reached. PNP-MAC also has a fixed CFP period; so, the throughput of PNP-MAC also becomes constant after the traffic load exceeds the number of CFP slots. As the proposed protocol contains a dynamic CFP period and also some packets are passed through the CAP period, the throughput of the proposed protocol continues to grow.

In Figure 7, we evaluate the power consumption of the compared protocols. Here, the IEEE 802.15.4 protocol shows a low consumption due to the long inactive period. The IEEE 802.15.4 contains a fixed CAP and a fixed CFP of 7 slots; so,

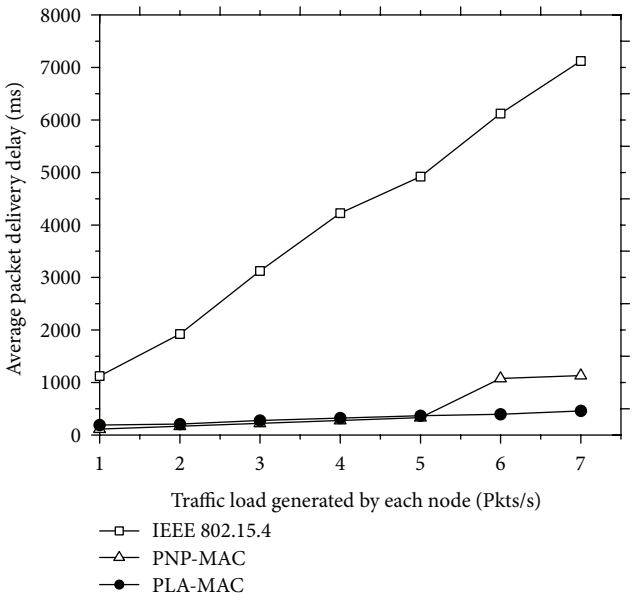


Figure 8: Average packet delivery delay versus traffic loads.

the power consumption is also fixed, regardless of the traffic load. The PNP-MAC protocol also has a fixed number of CFP periods, which is 40 in this simulation; so, the power consumption here is fixed as well. But the proposed protocol consists of a dynamic superframe structure that varies depending on the traffic load. So, the power consumption of the proposed protocol is low when the traffic is low, and it increases linearly with the traffic load.

5.3.2. Impacts of Traffic Load. Now, we measure the performance of PLA-MAC with respect to various traffic load. For

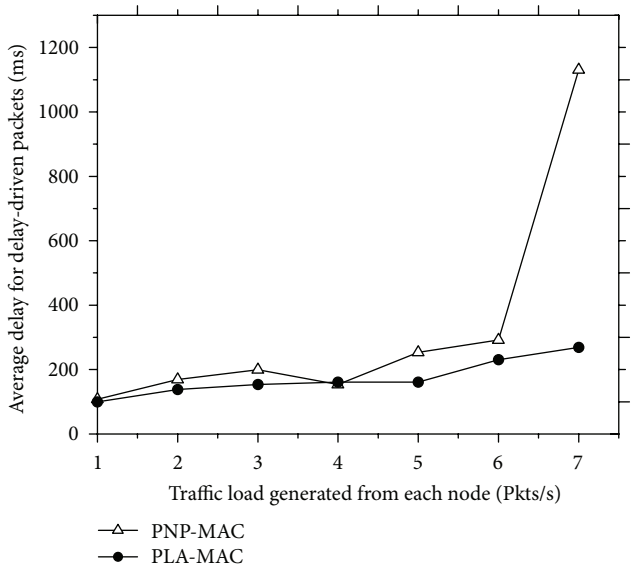


Figure 9: Average delay for delay-driven packet versus traffic load.

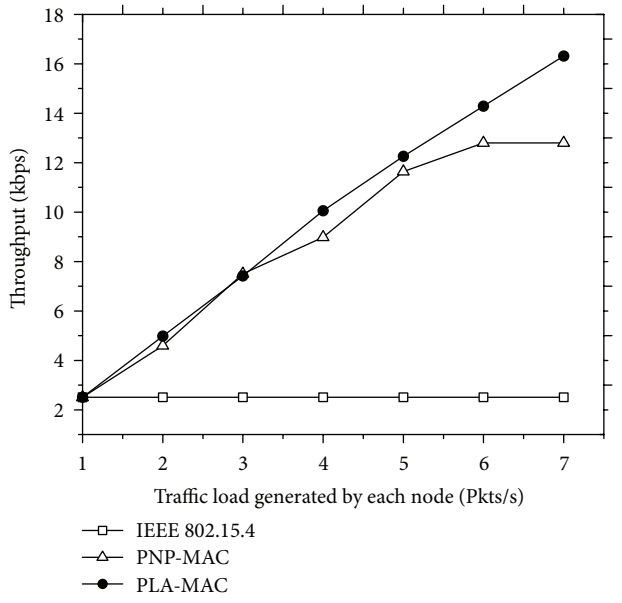


Figure 10: Throughput versus traffic load.

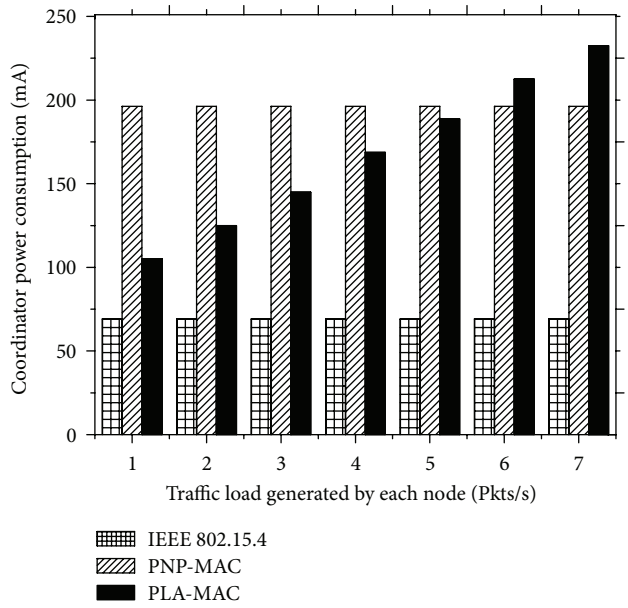


Figure 11: Coordinator power consumption versus traffic load.

this, we having are a fixed number of nodes which is seven, and the traffic load will vary from 1 Pkts/s to 7 Pkts/s.

In Figure 8, we perform the evaluation of average packet delivery delay with respect to diverse traffic load. We can see that PLA-MAC shows a good result compared to the other protocols specially in the higher traffic load section. The IEEE 802.15.4 experiences a large amount of delay, and the delay increases with the increasing traffic loads. The PNP-MAC protocol shows lower delay than the IEEE 802.15.4, but this delay is increased when traffic load is larger than the fixed number of GTS slots of PNP-MAC. The PLA-MAC is capable to achieve consistent low packet delivery delay with the increasing traffic loads. This nice result is the artifact of traffic load adaptive dynamic superframe structure and special treatment of DP packets defined in our proposed PLA-MAC.

In Figure 9, we can see the average delivery delay for the delay-driven packets. Here, PLA-MAC shows an overall low delay, but the PNP-MAC fails to do so. The PNP-MAC is a priority-based protocol, but it does not have any special scheme for the delay-driven packets; they are transmitted in the same way the other packet follows in the CFP periods. But in PLA-MAC, delay-driven packets enjoy a special service so that they can be delivered within their time constraint. In PLA-MAC, the delay-driven packets can be transmitted directly in the CAP period. So, the delay is low even in diverse traffic loads.

Next, we evaluate the throughput of the compared protocols in Figure 10. We can see that PLA-MAC achieves the maximum throughput of 16.3 kbps in the compared protocols. Here, as the IEEE 802.15.4 has only 7 GTS slots, so the throughput never increases from the first transmission. The growth of PNP-MAC also stops after the traffic load exceeds the fixed number of CFP slots. But because of the adaptive CFP period in PLA-MAC, the throughput of the proposed protocol continues to grow gradually.

In Figure 11, we evaluate the power consumption of the coordinator of the studied protocols. It presents similar results with that in Figure 7. Here, the IEEE 802.15.4 and the PNP-MAC protocol show a fixed power consumption, the reason being the fixed number of CFP slots in both protocols. But the PLA-MAC protocol shows a varying power consumption depending on the amount of traffic load, as it contains a dynamic superframe structure with adaptive CFP period.

In Figures 7 and 11, we observe that PLA-MAC consumes more power than IEEE 802.15.4 and PNP-MAC protocols when the number of source nodes is more than 8. Actually,

there is a tradeoff here between the network throughput and energy consumption of nodes. In Figures 6 and 10, we see that the achieved throughput of our proposed PLA-MAC protocol is better than the other two state-of-the-art protocols. Also, the average packet delay is much lower than the other two (Figures 4, 5, 8, and 9). However, the energy consumption of the PLA-MAC is little higher than the other two when the number of end devices is more than 8. This is happening due to carrying additional data packets compared to others. For example, our PLA-MAC consumes 10% additional energy compared to PNP-MAC when the number of end devices is 10 however, it achieves 22% better throughput performance.

6. Conclusion

In this paper, we have proposed an MAC protocol that provisions QoS to the packets according to their importance. The packets with higher priority get better service than the packets with lower priorities, which is very significant in medical applications of body area sensor network, as the higher priority packets may contain emergency data. The delay-driven packets are transmitted in the CAP period; so, they encounter minimum delay. The critical and reliability packets are transmitted in CFP period; so, reliability is also ensured. The superframe structure adapts its active section length according to the traffic load and the calculations for priority classification at the sensor nodes are kept to a minimum level; so, efficiency in power consumption is also maintained. Based on the simulation results, we can state that our PLA-MAC significantly improves the QoS performances due to its judicious transmission scheduling according to traffic prioritization and load adaptiveness.

Acknowledgment

This work was supported by the Research Center of College of Computer and Information Sciences, King Saud University, Riyadh, Kingdom of Saudi Arabia. The authors are grateful for this support.

References

[1] M. A. Ameen, A. Nessa, and K. S. Kwak, "QoS issues with focus on wireless body area networks," in *Proceedings of the 2008 3rd International Conference on Convergence and Hybrid Information Technology (ICCIT '08)*, pp. 801–807, IEEE Computer Society, Washington, DC, USA, 2008.

[2] B. Latré, B. Braem, I. Moerman, C. Blondia, and P. Demeester, "A survey on wireless body area networks," *Wireless Networks*, vol. 17, no. 1, pp. 1–18, 2011.

[3] A. D. Wood, J. A. Stankovic, G. Virone et al., "Context-aware wireless sensor networks for assisted living and residential monitoring," *IEEE Network*, vol. 22, no. 4, pp. 26–33, 2008.

[4] M. A. Hanson, H. C. Powell, A. T. Barth et al., "Body area sensor networks: challenges and opportunities," *Computer*, vol. 42, no. 1, pp. 58–65, 2009.

[5] "NIST," http://www.antd.nist.gov/ban/ .

[6] G. Fang and E. Dutkiewicz, "BodyMAC: energy efficient TDMA-based MAC protocol for wireless body area networks," in *Proceedings of the 9th International Symposium on Communications and Information Technology (ISCIT '09)*, pp. 1455–1459, September 2009.

[7] S. Marinkovic, C. Spagnol, and E. Popovici, "Energy-efficient TDMA-based MAC protocol for wireless body area networks," in *Proceedings of the 3rd International Conference on Sensor Technologies and Applications (SENSORCOMM '09)*, pp. 604–609, June 2009.

[8] D. Cavalcanti, R. Schmitt, and A. Soomro, "Performance analysis of 802. 15. 4 and 802. 11e for body sensor network applications," in *Proceedings of the International Federation for Medical and Biological Engineering (IFMBE '07)*, vol. 13, 1st Session, pp. 9–14, 2007.

[9] B. Otal, L. Alonso, and C. Verikoukis, "Novel QoS scheduling and energysaving MAC protocol for body sensor networks optimization," in *Proceedings of the ICST 3rd International Conference on Body Area Networks (BodyNets '08)*, article 27, 2008.

[10] IEEE Standartd for Information Technology 802.15.4, *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*, Institute of Electrical and Electronics Engineers, New York, NY, USA, 2003.

[11] N. F. Timmons and W. G. Scanlon, "Analysis of the performance of IEEE 802.15.4 for medical sensor body area networking," in *Proceedings of the 1st Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (IEEE SECON '04)*, pp. 16–24, October 2004.

[12] "IEEE 802.15 WPAN Task Group 4 (TG4)," http://www.ieee802.org/15/pub/TG4.html.

[13] G. Zhou, J. Lu, C. Y. Wan, M. D. Yarvis, and J. A. Stankovic, "BodyQoS: adaptive and radio-agnostic QoS for body sensor networks," in *Proceedings of the 27th IEEE Communications Society Conference on Computer Communications(INFOCOM '08)*, pp. 1238–1246, April 2008.

[14] O. Md. Rahman, C. S. Hong, S. Lee, and Y.-C. Bang, "ATLAS: a traffic load aware sensor MAC design for collaborative body area sensor networks," *Sensors*, vol. 11, no. 12, pp. 11560–11580, 2011.

[15] J. S. Yoon, G. S. Ahn, S. S. Joo, and M. J. Lee, "PNP-MAC: preemptive slot allocation and non-preemptive transmission for providing QoS in body area networks," in *Proceedings of the 7th IEEE Consumer Communications and Networking Conference (CCNC '10)*, January 2010.

[16] C. Li, B. Hao, K. Zhang, Y. Liu, and J. Li, "A Novel Medium Access Control Protocol with Low Delay and Traffic Adaptivity for Wireless Body Area Networks," *Journal of Medical Systems*, pp. 1–11, 2011.

[17] A. Razzaque, C. S. Hong, and S. Lee, "Data-centric multiobjective QoS-aware routing protocol for body sensor networks," *Sensors*, vol. 11, no. 1, pp. 917–937, 2011.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 974705, 7 pages
http://dx.doi.org/10.1155/2013/974705

Research Article

A Mobility-Aware Efficient Routing Scheme for Mobile Sensor Networks

Jinman Jung,[1] Yookun Cho,[1] and Jiman Hong[2]

[1] *School of Computer Science and Engineering, Seoul National University, Seoul 151-744, Republic of Korea*
[2] *School of Computing, Soongsil University, Seoul 156-743, Republic of Korea*

Correspondence should be addressed to Jiman Hong; jiman@ssu.ac.kr

Received 10 December 2012; Accepted 20 February 2013

Academic Editor: Wei Wang

In mobile sensor networks, a variety of dynamic environmental conditions affect performance. Node movement caused by the network environment or by mobile entities makes the sensors particularly vulnerable to route failures, which in turn affects the efficiency and reliability of these networks. Therefore, mobility is an important factor in the design of a routing protocol for mobile sensor networks. In this paper, we propose a mobility-aware efficient routing called MAER, in which sensor nodes make use of mobile information to select the most appropriate routing behavior. The proposed method integrates proactive and reactive routing components efficiently using a sink cluster that consists of underlying multiple static or slow sensor nodes. The cluster provides the stable paths between less mobile entities efficiently. Our scheme also uses mobile information to evaluate and select alternative paths during a route discovery process, thus allowing sensor nodes dynamically to adapt to varying mobile networks. We evaluate the performance of MAER using a simulation by comparing it to the most popular standard for WSN, AODV. The results of the simulation show that our scheme outperforms AODV in mixed mobile sensor networks.

1. Introduction

Mobility sensor networks (MSNs) are increasingly emerging in various application domains, from environmental monitoring to intelligent industrial automation. In MSNs, the energy efficiency during routing operations has been a primary metric because typical sensor nodes have constrained battery power [1]. However, the mobility that results from network environmental influences (e.g., wind and water) or from mobile objects (e.g., human, animal, robot, and vehicles) can degrade the energy efficiency of sensors significantly [2]. Also, as mobility increases, routing schemes are affected in different ways. In proactive schemes such as DSDV [3], the number of control messages increases dramatically to maintain the route before the link breaks regardless of whether data is being transmitted. On the other hand, in reactive schemes such as AODV [4], the sensor nodes discover the route to the destination only if they are needed. These schemes can also reduce the control overhead compared to proactive approaches. However, more frequent changes of the topology can lead to a failure of the discovery process even under a reactive scheme. Thus, we need to design a more efficient and robust routing scheme suitable for MSNs.

We focus on a hybrid approach that relies on mobility information and assumes that each sensor has different mobility information in a mixed sensor network consisting of static and mobile nodes although all sensors have the same mobility model. Most hybrid approaches concentrate on traffic loads considering the impact of the locality. However, in highly mobile sensor networks, the impact of mobility is more important than the benefits of the locality because most undelivered messages are caused by broken links due to node mobility, and the message loss can lead to numerous control messages from frequent route-maintenance and repetitive route-discovery attempts. In order to reduce the control overhead, our approach utilizes a stable path that is established between less mobile entities through the use of mobility information.

In this paper, we propose a mobility-aware efficient routing scheme called MAER, in which sensor nodes make use of their mobile information to select the most appropriate

routing behavior and the best path among candidates. This represents a new integrated approach which includes both proactive and reactive strategies and hence is designed to work efficiently under varying mobile networks. The proposed routing scheme is based on a sink cluster. It takes into account the mobile status of each sensor node during clustering. A cluster consists of a spanning tree with irregular boundaries which relies on the number of static or slow nodes, unlike a traditional cluster, the size of which is the maximum number of hops from a cluster head. This modification lowers the cost of MAER by minimizing reconstructions of clusters. In particular, it adapts to varying mobile sensor networks by shrinking or expanding a cluster. Slow nodes in the cluster perform proactive routing by updating and recovering the routes within the cluster. Meanwhile, mobile nodes that do not participate in the cluster perform the reactive routing part. Furthermore, MAER uses an improved mobility-aware query control mechanism based on a sink cluster.

The rest of this paper is organized as follows. Section 2 addresses related works. Section 3 introduces the sink cluster that is established between less mobile entities for mobile sensor networks. The proposed mobility-aware efficient routing protocol is then given. Section 4 presents the simulation and experimental results. Finally, Section 5 summarizes this work.

2. Related Works

Routing protocols are classified into two categories: proactive (e.g., DSDV [3], OLSR [5], SPIN [6], and OSPF [7]) and reactive (e.g., AODV [4], DSR [8], LAR [9], and directed diffusion [10]). In proactive scheme, each node propagates control messages to maintain fresh entries in the routing table when the network topology changes. For example, in OSPF [7], sensors maintain routes proactively by depending on soft timers to detect a link failure. If a neighbor entry in route table is not periodically refreshed by a HELLO message, the entry is invalidated. Meanwhile, in reactive schemes, nodes send control messages to find the path to the destination only if there is data to send. The most popular standard for WSN is the IEEE 802.15.4 ZigBee standard, where a ZigBee network layer uses an ad hoc on-demand distance vector (AODV) routing [4]. In AODV, if there is no cached entry in the routing table, the source node initiates a route discovery process by broadcasting a route request (RREQ) message, which is rebroadcasted by other nodes until it reaches its destination. The destination node then responds via a route-reply (RREP) message which is sent back to the source node via a unicast reply mechanism. Directed diffusion [10] is another reactive routing scheme for WSNs. The sink propagates an interest message when it needs to collect data from sensor nodes. A proactive scheme can result in better performance with static networks in terms of latency because the routing information is maintained at all times. In contrast, a reactive scheme can perform more efficiently than a proactive scheme when there is greater mobility.

Hybrid schemes which combine proactive and reactive schemes have also been proposed. Examples include ZRP [11], EAGER [12], MSA [13], and ARPM [14]. The Zone Routing Protocol (ZRP) [11], in which each node has its own proactive zone with a proactive scheme performed within each zone, was the first hybrid routing scheme. Routing between nodes in different zones is performed by a reactive routing scheme. In EAGER [12], the network is partitioned into disjointed proactive cells in which each node utilizes a proactive scheme for intracell routings and reactive routing for intercell routings. The optimal cell size and transmission range are obtained analytically. In MSA [13], the authors propose a framework to select the best routing strategy given the mobility of the source and destination. They incorporate existing routing protocols into the framework. In contrast, ARPM [14] redesigns a hybrid protocol by integrating DSDV and AODV as a proactive and a reactive scheme, respectively. In ARPM, each node utilizes a different type of routing depending on the node's mobility.

The proposed MAER protocol differs from traditional zone- or cell-based adaptive schemes in terms of its network organization. In MAER, a cluster has a spanning tree with irregular boundaries made up of a number of less mobile entities so as to provide a stable path. The cluster in MAER adapt to varying mobile sensor networks by shrinking or expanding.

3. MAER: Mobility-Aware Efficient Routing for Mobile Sensor Networks

In this section, we present MAER, a mobility-aware efficient routing scheme for mobile sensor networks. First, we introduce the concept of a sink cluster in a mobile sensor network, after which we describe how MAER works in varying mobile sensor networks.

3.1. Sink Cluster. We take into consideration a network with N randomly dispersed mobile sensor nodes (MS_i, $0 \leq i \leq N$) and a sink node. We assume that all of the nodes have the same wireless transmission range r. Two nodes are considered neighbors if they are within the transmission range of one another. The nodes are moving randomly in different directions at different speeds. Each MS_i contains the mobility information of the speed, which is measured at the node. To handle link failures and provide stable paths efficiently, MAER maintains a sink cluster rooted at the sink. Sensor nodes decide to join or leave a sink cluster based on their own mobile information relative to that of other members of the cluster.

A cluster has a spanning tree formed by slow sensor nodes with low levels of mobility (as characterized by, for instance, slow speeds and large pause times) under a given threshold. Let us consider a cluster called sink cluster (SC). It is formed by associating the sink with multiple slow nodes of height h. SC is composed of SCMs and a sink where SCM is a sink cluster member. The SCMs exploit a proactive routing protocol while also playing an intermediary role in

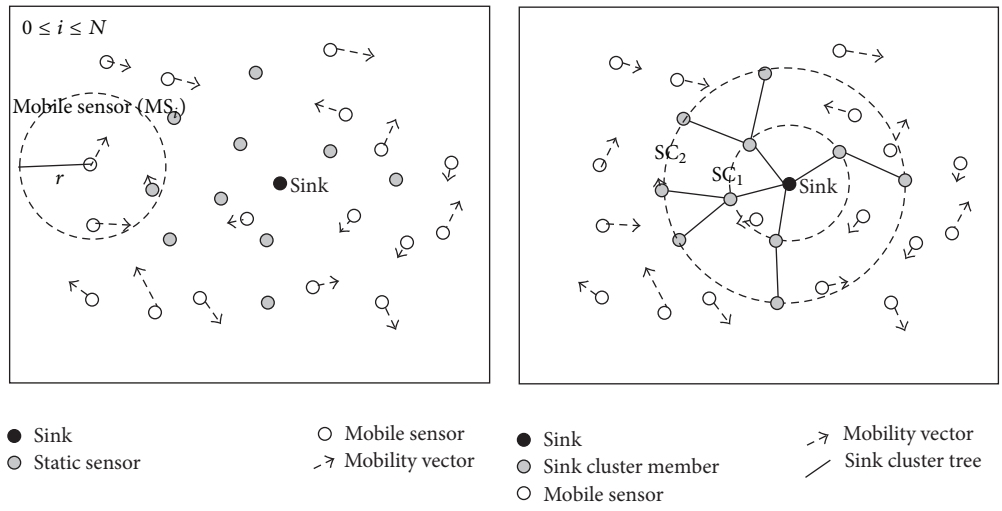


Figure 1: (a) Mobile sensor networks and (b) a sink cluster for $h = 2$.

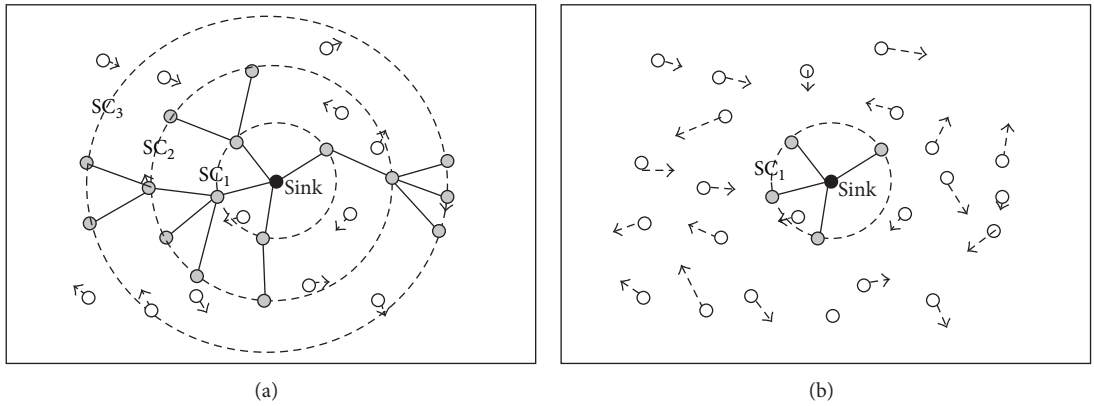


Figure 2: The (a) expansion and (b) shrinking of a sink cluster according to the degree of node mobility.

efficient route discovery mechanisms, as described in the next subsection. We denote a set of nodes which can reach the sink in k hops as SC_k ($0 \le k \le h$). Accordingly, the sink node belongs to SC_0.

The tree organization of an SC is initiated by the sink. The sink creates the cluster by periodically sending an *advertisement* broadcast message. A sensor node that receives the *advertisement* message within a single hop sends a *join* message to the sink if its degree of mobility drops below a given threshold λ_x. In response to the *join* message, an *acknowledgement* message is sent from the sender of the *advertisement* message. Upon the reception of the *acknowledgement* message, the sensor node becomes the SCM node and registers the sender of the message as a parent node with a distance vector from the sink. Consequently, all of the neighbors that have mobility less than λ_x join the cluster as an SC_1 node. Also, each SCM periodically sends *advertisement* broadcast messages during every interval T_a, as our objective is to group all slow nodes for a more stable path. In this way, a sensor node joins SC_k and becomes an SCM if the sensor node receives an *advertisement* message from an SCM in SC_{k-1} within a single hop and if the node's mobility is below a given threshold λ_x. SCMs can also change their own parent node if the distance vector from other *advertisement* messages is less than their current distance vector. When the mobility of a SCM exceeds the threshold λ_y, the node leaves the sink cluster by sending a *leave* message to both its parent and child nodes. If its parent node is no longer a SCM, the child node also leaves the sink cluster. Figures 1(a) and 1(b) illustrate mobile sensor networks and an example of a sink cluster when $h = 2$, respectively.

This approach helps to minimize clustering overhead because the SC is composed of relatively slow nodes. In addition, the sink cluster topology is adaptive to varying networks. The cluster grows automatically when static or slow nodes are greater in number in the network and shrinks if more sensor nodes are moving faster, as shown in Figure 2. Furthermore, the sink cluster enables MAER to provide more efficient routing by integrating proactive and reactive routing schemes, as described in the next subsection.

3.2. Integrating of Proactive and Reactive Routing Components. In this subsection, we extend the discussion to the area outside of the sink cluster and present the details of the key design of integration of the reactive routing component based on the sink cluster. The SCM nodes perform proactive routing within the cluster infrastructure while the other mobile sensors use the reactive routing routine. The goal of reactive routing here is efficiently to find alternative stable paths for mobile nodes that do not belong to the cluster.

Conventional reactive schemes typically use a rebroadcast mechanism to accomplish route discovery, where every node broadcasts upon receiving a route request (RREQ) message for the first time. The rebroadcast mechanism, however, can lead to high channel contention and collisions, often known as the rebroadcast storm problem [15]. In MAER, because the SCM nodes already have a destination route, they stop rebroadcasting when a RREQ message is received. Instead, the SCM nodes convert the broadcast-RREQ message into a unicast-RREQ message and forward the converted message to their parent SCM node. With the help of the query conversion mechanism, MAER becomes more efficient than the conventional route discovery process.

We also enhance the route discovery process by the mobility factor for path selection. In MAER, each MS_i has a weight value, denoted as ϕ_i, assigned as a cost according to its own mobile level. The mobility factor is the cumulative sum of the weight values of all intermediate nodes that form the path. This approach requires the modification of the routing table so that it contains the mobility factor for the paths. When a RREQ message arrives, each node adds its weight value to the mobility factor field of the message and the node updates its routing table. This procedure is repeated until the RREQ message reaches its destination. The cumulative mobility factor included in the RREQ message is recorded in the destination's route table. In addition, the expected mobility factor for a path with l hops can be calculated by dividing the factor by the hop count. Note that a lower mobility factor indicates that there are fewer mobile entities among the intermediate nodes in the path, whereas a higher mobility factor implies the opposite. As a result, the path with the minimum mobility factor is utilized during the route discovery process instead of the shortest path. Although the path may not the shortest, it can be a more stable path when established between less mobile entities. Therefore, when RREQ messages arrive, each node has to update the route table if the mobility factor in the RREQ message is lower than that in the route table. The SCM nodes, however, have to forward the unicast RREQ to their parent node after updating the routing table entry. When the RREQ reaches its destination, a route reply (RREP) message is not immediately sent to the source node. To construct a stable route, the RREP message is delayed for an acceptable time. However, this lazy reply mechanism can lead to long setup times when discovering routes with lower mobility factors.

To illustrate how MAER routing works, Figure 3 shows an example with a sink cluster. When MS_1 has sensed data to send to the sink but has no existing route to the sink, it initiates a route discovery process by broadcasting a RREQ message that contains its weight value ϕ_1 of 3 for

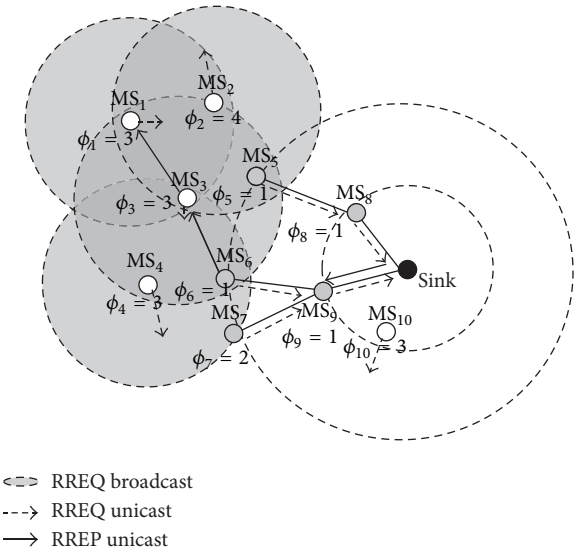


Figure 3: An example of our mobility-aware efficient routing.

the mobility factor. MS_2 and MS_3, upon receiving the RREQ for the first time, add their weight values to the mobility factor in an accumulative manner and then rebroadcast the message. Then, MS_4, MS_5, and MS_6 repeat this rebroadcast procedure. Nodes MS_1, MS_2, MS_3, and MS_4 can receive several RREQs, but they do not forward it again because they have already forwarded the RREQ once. Instead, they compare the mobility factor to their current mobility factor in the route table entry in order to determine better paths for every RREQ arrival time. Because MS_5, MS_6, and MS_7 are SCM nodes that participate in the sink cluster, they convert the RREQ into the unicast type and forward it to the parent nodes MS_8, MS_9, and MS_9, respectively; specifically, MS_5 does not convert the RREQ message received from MS_2 to the parent node MS_8 because the mobility factor of $(\phi_1 + \phi_2)$ is higher than the current mobility factor of $(\phi_1 + \phi_3)$ in the route table. Likewise, MS_6 does not forward the RREQ message received from MS_4 again. MS_9 also does not send the message from MS_7 to its parent node again. When the sink receives the RREQ, the sink initiates the lazy reply process. After waiting time T_s, the destination node creates the RREP and sends it back. Although the RREQ from MS_8 arrives first, the sink selects MS_9 for the next node toward the source node.

4. Performance Evaluation

In this section, we evaluate the performance of MAER. We compare the proposed mechanism of MAER with one of most popular standard routing protocols for mobile sensor networks, AODV. We also use two different advertisement intervals (T_a = 0.5 and 2) in MAER. For join and leave messages, the thresholds λ_x and λ_y are given values of 4 m/s and 8 m/s, respectively. We assigned a weight value to each of four levels with a preliminary cost of 4, 3, 2, or 1 according to the mobility threshold, which have values of 4 m/s, 8 m/s, and 16 m/s. In all simulations, 100 sensor nodes are randomly

Table 1: Parameters used in the simulation.

Notation	Description
Size of field	$500 \times 500\ m^2$
Distribution of nodes	Random distribution
Number of nodes (N)	100
Transmission range (r)	100
Data packet size	256 bytes
Mobility model	RDMM (Random direction mobility model)
Traffic load	Constant bit rate
Propagation model	Free space model

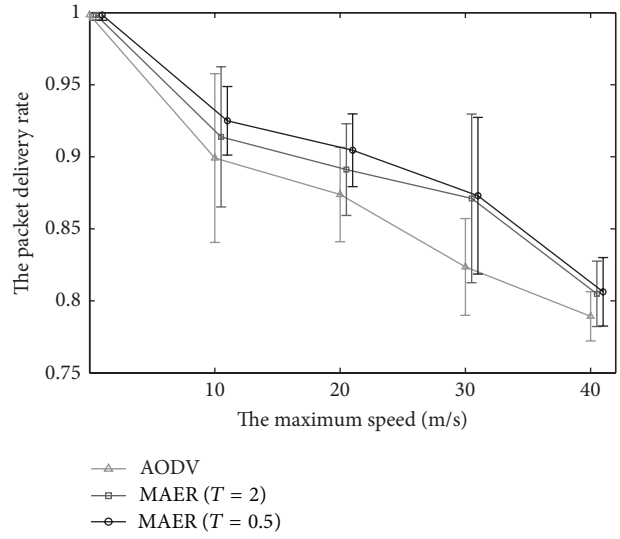


Figure 4: The packet delivery rate according to the mobility of node.

deployed in a square area with a size of 500 m × 500 m. After the initial placement, nodes move continuously according to the RDMM [16] model; every node keeps moving in a random direction at a random speed during every epoch. Each node has the same transmit power of coverage of 100 m. We also assume that the channels are free of errors and collisions. Each data is averaged over 10 simulation runs with a different seed. Based on the parameters described above, we consider two different mobility scenarios: (i) a uniform case in which all nodes keep moving continuously according to the same RDMM model and (ii) a mixed case where the nodes are categorized as either static nodes or mobile nodes according to their RDMM behavior.

We consider the packet delivery rate and the control overhead in the following two-part metric: (1) the ratio of the total number of packets that the sink receives to the total number of packets that the sensor nodes send and (2) the number of control bytes for reception and transmission. We evaluated this metric according to various maximum node speeds and numbers of static nodes for each model. Table 1 shows some of the important simulation parameters.

Figure 4 shows a comparison of the average packet delivery rate with respect to different degrees of mobility

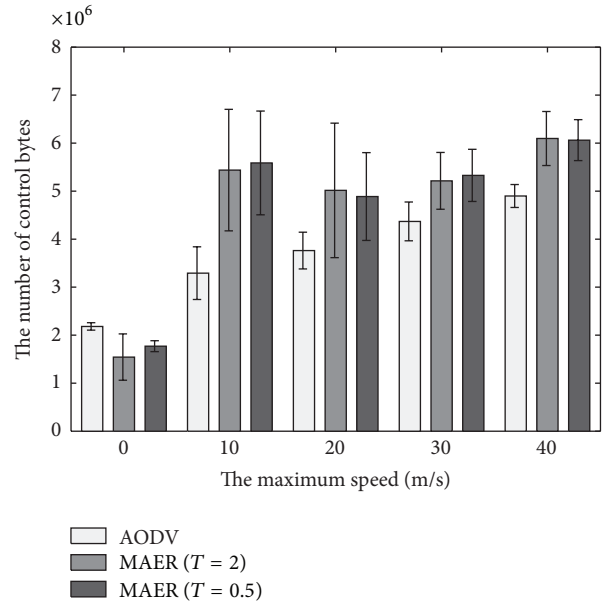


Figure 5: The control overhead according to the mobility of node.

in the uniform mobility case. In all schemes, the packet delivery rate decreases as the maximum speed of the nodes increases from 10 m/s to 40 m/s because the broken links caused by the mobility characteristic degrade the packet delivery ratio; specifically, we found that the conventional AODV scheme has increased difficulty when dealing with high node mobility. MAER improves the packet delivery ratio by about 5% compared to AODV. As expected, MAER can be improved with a shorter interval in terms of the packet delivery rate, as the advertisement messages stabilize the clusters. Figure 5 shows the number of control overhead bytes needed to achieve the packet delivery. The control overhead of AODV is more significant when the mobility of the nodes increases. Due to its frequent messages, MAER usually generated more control traffic than AODV when the maximum speed equaled or exceeded 10 m/s. The control overhead of MAER did not scale linearly with high mobility.

Figure 6 shows the packet delivery rate for different number of static nodes in the mixed case (i.e., a mixed MSN consisting of both static and mobile nodes). The packet delivery rate of MAER is better than that of AODV. In particular, we achieved a significantly higher packet delivery rate at a lower cost in the mixed case. Moreover, MAER yields better results for high mobility. This improvement is attributed to the static nodes used to provide stable paths from among alternatives regardless of the maximum speeds of mobile node.

Figure 7 shows the control overhead with various static nodes. Here, AODV appears to perform best. However, when taking into account the packet delivery ratio, as shown in Figure 6, AODV shows the worst result. As the number of static nodes increases, the difference of control bytes between AODV and MAER becomes smaller. This is largely due to the fact that they adapt dynamically to network conditions

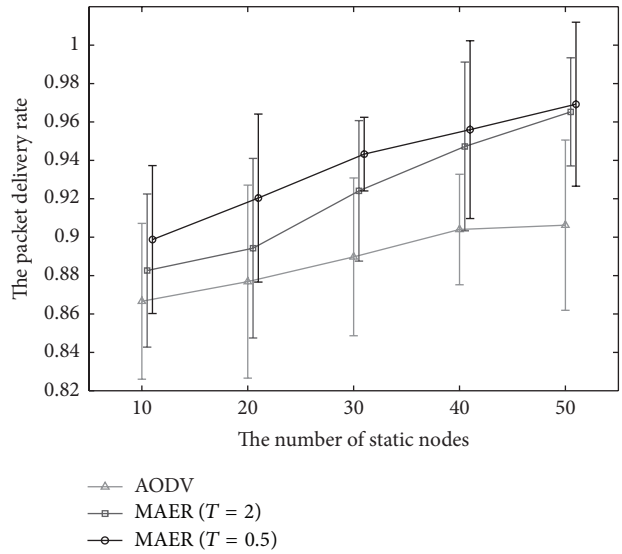


Figure 6: The packet delivery rate according to the number of static nodes.

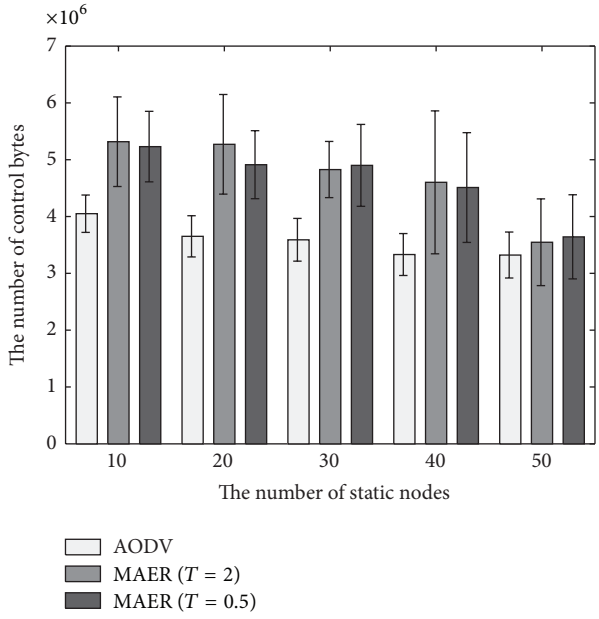


Figure 7: The control overhead according to the number of static nodes.

through an efficient discovery process based on the sink cluster.

For a fair comparison, we measured the number of control bytes per data packet delivered. Figure 8 shows the control overhead per successful data transmission according to the traffic load. The simulation results show that MAER outperforms AODV in all cases; specifically, when the traffic load is heavier (packet traffic rate ≤2), MAER with $T_a = 0.5$ generates relatively less control overhead compared to the case when $T_a = 2$. This indicates that the effect of updating the cluster becomes less efficient under a light traffic load.

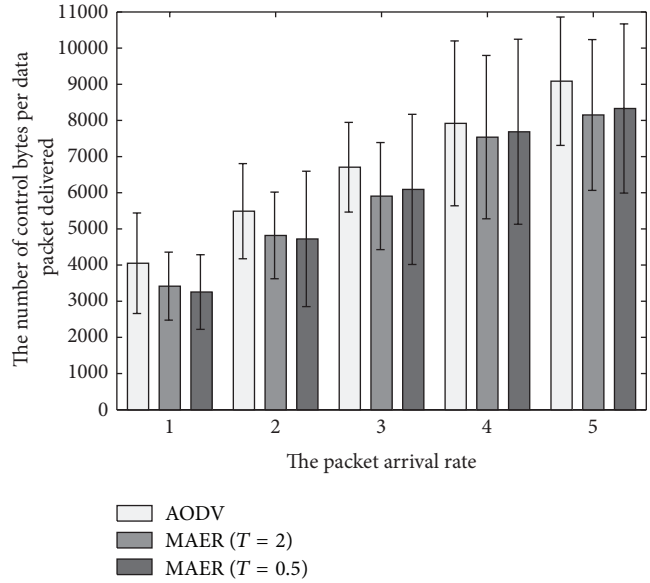


Figure 8: The control overhead per successful data transmission.

5. Conclusions

We have presented a mobility-aware routing scheme for mobile sensor networks, called MAER. The goal of MAER is to adapt to mobile sensor networks by taking into consideration the node's level of mobility. Our scheme exploits a sink cluster formed by slow nodes with low mobility to establish a stable path between less mobile entities. It also provides stable paths in an efficient manner using a query control mechanism based on the mobility factor. Simulation results show that our scheme outperforms AODV in terms of the packet delivery rate and control overhead in mixed mobile sensor networks.

References

[1] J. Lian, K. Naik, and G. B. Agnew, "Data capacity improvement of wireless sensor networks using non-uniform sensor distribution," *International Journal of Distributed Sensor Networks*, vol. 2, no. 2, pp. 121–145, 2006.

[2] K. Ma, Y. Zhang, and W. Trappe, "Managing the mobility of a mobile sensor network using network dynamics," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 1, pp. 106–120, 2008.

[3] C. E. Perkins and P. Bhagwat, "Highly dynamic destination sequenced distance vector routing (DSDV) for mobile computers," in *Proceedings of the ACM Conference on Communications Architectures, Protocols and Applications (SIGCOMM '94)*, pp. 234–244, 1994.

[4] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector(AODV) routing," Tech. Rep. 3561, IETF MANET Working Group, 2003.

[5] J. Kulik, W. Heinzelman, and H. Balakrishnan, "Negotiation-based protocols for disseminating information in wireless sensor networks," *Wireless Networks*, vol. 8, no. 2-3, pp. 169–185, 2002.

[6] T. H. Clausen and P. Jacquet, RFC3626, Internet Engineering Task Force (IETF), 2003, http://www.ietf.org/rfc/rfc3626.txt.

[7] J. Moy, OSPF version 2 RFC, 2328, 1998.

[8] D. Johnson, D. Maltz, and Y. C. Hu, "The dynamic source routing protocol for mobile ad hoc networks (DSR)," Tech. Rep., IETF MANET Working Group, 2004.

[9] Y.-B. Ko and N. H. Vaidya, "Location-aided routing (lar) in mobile ad hoc networks," Tech. Rep., Department of Computer Science, Texas A&M University, College Station, Tex, USA, 1998.

[10] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed diffusion for wireless sensor networking," *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 2–16, 2003.

[11] Z. J. Haas and M. R. Pearlman, "The Zone Routing Protocol (ZRP) for ad hoc networks," Tech. Rep., IETF, Internet Draft, draft-zone-routing-protocol-02.txt, 1999.

[12] Q. Zhao, L. Tong, and D. Counsil, "Energy-aware adaptive routing for large-scale ad hoc networks: protocol and performance analysis," *IEEE Transactions on Mobile Computing*, vol. 6, no. 9, pp. 1048–1059, 2007.

[13] A. Bamis, A. Boukerche, I. Chatzigiannakis, and S. Nikoletseas, "A mobility aware protocol synthesis for efficient routing in ad hoc mobile networks," *Computer Networks*, vol. 52, no. 1, pp. 130–154, 2008.

[14] H. L. Seba, "ARPM: an adaptive routing protocol for MANETs," in *Proceedings of the International Conference on Pervasive Services (ICPS '06)*, pp. 295–298, June 2006.

[15] S. Ni, Y. Tseng, Y. Chen, and J. Sheu, "The broadcast storm problem in a mobile ad hoc Network," in *Proceedings of the 5th ACM/IEEE Annual International Conference on Mobile Computing and Networks (MOBICOM '99)*, pp. 151–162, 1999.

[16] E. M. Royer, P. M. Melliar-Smith, and L. E. Moser, "An analysis of the optimum node density for ad hoc mobile networks," in *Proceedings of the IEEE International Conference on Communications (ICC '01)*, pp. 857–861, June 2001.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 324641, 20 pages
http://dx.doi.org/10.1155/2013/324641

Research Article

Providing Virtual Memory Support for Sensor Networks with Mass Data Processing

Nan Lin,[1] Yabo Dong,[1,2] and Dongming Lu[1,2]

[1] *School of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China*
[2] *Cyrus Tang Center for Sensor Materials and Applications, Zhejiang University, Hangzhou 310027, China*

Correspondence should be addressed to Nan Lin; nlin@zju.edu.cn

Received 26 October 2012; Accepted 28 January 2013

Academic Editor: Sheikh I. Ahamed

With the development of sensor networks and emerging of various sensors, sensor networks are capable of acquiring mass data to achieve much more complex monitoring tasks than ever. For example, image sensor nodes take photos using cameras, and images are collected and processed or stored for further processing. So, mass data processing is required for these sensor networks. However, low-power resource-constrained sensor nodes are normally equipped with kilobytes of RAM which might be not enough for storing large data for processing. In this paper, we propose an optimized virtual memory mechanism for large data processing on low-power sensor nodes. We point out the major overhead of virtual memory for large data processing on sensor nodes and introduce efficient solutions to address these issues. Evaluation shows that the overhead of the proposed virtual memory is reduced to an affordable range. We further compare the energy consumption of data processing programs using virtual memory with other means that process or transmit data. Data processing using virtual memory can be significantly more energy efficient than data processing using rich-resource sensor nodes or transmitting data to powerful gateways for central processing.

1. Introduction

Sensor networks are used in various areas nowadays, like environment monitoring, agriculture, video surveillance, and so forth. Sensor nodes now collect not only scalar sensed data from the environment, but also stream of mass data, for example, videos and images. Fast in situ processing of large data might be required to enable fast response or to reduce transmission overhead. Sensor nodes are expected to carry out large data processing with reasonable overhead. However, low-power sensor nodes are known for their limited resources. For instance, motes are equipped with kilobytes of RAM which may easily be insufficient for storing or processing images. Programs for large data processing on sensor nodes may require much more memory than the RAM size of the microcontroller.

There would generally be three different methods to address this problem listed below.

(1) Use more powerful microcontrollers with sufficient RAM for large data processing.

(2) Employ virtual memory to increase the available memory for data processing programs.

(3) Transmit large data to powerful gateway for central processing.

Using more powerful microcontrollers equipped with sufficient RAM for data processing would be a straightforward solution for large data processing. For example, Intel Mote 2 is a sensor platform with increased CPU performance and improved radio bandwidth to acquire, process, and transmit large data streams [1]. The platform provides 32 MB SDRAM on-board, which would be large enough for storing images or videos. Unfortunately, SDRAM has been known to be more expensive and energy consuming per bit compared to other memory mediums (NOR flash, NAND flash, etc.). This has led to the relative high power consumption of Intel Mote 2 in deep sleep mode. The current in deep sleep mode of Intel Mote 2 is about 387 μA which is much larger than that of traditional sensor nodes (about 10 μA) (e.g., MicaZ, TelosB). This would greatly reduce the lifetimes of the sensor nodes since sensor nodes are expected to be in deep sleep mode

most of time. So, sensor platforms with large RAM space may not be suitable for long-term surveillance.

Virtual memory [2] is generally to provide flat isolated memory address spaces for programs on conventional computer architectures. With the support of Memory Management Unit (MMU), page faults can be caught and out-of-bound memory accesses can be protected from destroying other programs or the operating systems. However, virtual memory is generally used on resource-constrained sensor nodes to expand its limited RAM space to provide enough memory for complex programs, for example, TinyDB [3], whose memory footprint outruns the physical RAM provided by low-power microcontrollers. This is different from conventional virtual memory in that isolating memory errors of programs from destroying other parts of system is not a major concern. Without MMU, virtual memory cannot be implemented on low-power sensor nodes easily. Software-aided mechanisms must be used to allow transparent virtual memory access. Generally, assembly instructions accessing memory are replaced by snippets which redirect the memory accessing to read/write virtual memory. Despite the significant benefits of virtual memory, traditional sensor node programs do not employ virtual memory for its high overhead.

Traditionally, sensor nodes collect data and transmit data to central gateway for processing because sensor nodes are generally supposed to be resource constrained. Transmitting large data to gateway can be an option for sensor nodes if data processing is limited due to insufficient RAM. However, transmitting large data to gateway would incur much overhead because network transmission is known to be about 1000 times more energy hungry than data processing. To the best of our knowledge, there is currently no work to compare the energy consumption of network transmission and data processing using virtual memory in sensor networks.

We have been working on building an image sensor network in which sensor nodes capture images of the target object and do in situ image comparing to compress similar images to minimize network traffic. We have developed an image sensor node platform. STM32F103ZE [4] is used as the microcontroller whose deep sleep mode current is about 20 μA, making our image sensor node platform suitable for long-term surveillance. STM32F103ZE is equipped with 64 kilobytes of RAM. This is much larger than that of ATMega128 [5] and MSP430 [6], however, still far from enough for processing images taken from cameras. Without enough memory, algorithms and existing programs must be altered to work with limited memory. For example, image operations can be carried out block by block, intermediate results can be stored on external storage, and the final result can be calculated from the intermediate results. This might bring heavy overhead to the development of image sensor node programs. Our primal motivation for developing virtual memory is to support large data processing and enable long-term operation on sensor nodes at the same time. By developing a virtual memory mechanism for the Cortex-M3 [7] platform, existing algorithm implementations can be used directly without adaptations. We tried to provide megabytes of memory for programs to manipulate images all in memory. The image sensor nodes are equipped with NAND flash of size 2 gigabytes, and we use parallel NAND flash as the secondary storage media for the virtual memory. It is generally a valid assumption that image sensor nodes are equipped with large NAND flashes, mainly for storing sensor data.

The virtual memory mechanism is verified on our image sensor nodes using various general data processing programs. Evaluation shows that the overhead of virtual memory can be reduced dramatically using these optimizations. Our contribution lies in the fact that we prove that virtual memory can be used for sensor nodes to achieve more energy-efficient data processing than using high performance energy-consuming microcontrollers or transmitting large data to resourceful gateway and we have proposed a flash translation layer [8] to use NAND flashes efficiently for secondary storage accessing.

Rest of paper is organized as follows. We describe the challenges we have faced when developing the virtual memory mechanism in Section 2. Section 3 gives an introduction of related work. Section 4 gives an overview of the proposed virtual memory system. Section 5 describes the details of C code virtualization, in which last-cache buffers are used to accelerate address translations. Section 6 describes how NAND flash is used as secondary storage and the design and details of Lavish-FTL. Section 7 evaluates the overhead of the proposed virtual memory mechanism with data processing programs and compares the energy consumption of these programs with other options. Conclusion and future work is drawn in Section 8.

2. Challenges

Two major issues have been met when we were trying to make the virtual memory implementation efficient enough to be affordable for data processing programs. The two issues are discussed in the following two subsections. Section 2.3 gives our solutions for these two issues.

2.1. Code Virtualization. One requirement of the virtual memory is that programmers develop their programs unawares of the underlying virtual memory. Code virtualization is used to make programs work with virtual memory on MMU-less microcontrollers. For example, assembly instructions which access memory (i.e., LDR, STR in the ARM instruction set) can be replaced by routines which access virtual memory. We assume a flat address space provided by the virtual memory, mapped linearly to the secondary storage. It is reasonable when the size of the secondary storage is much larger than the required virtual memory size. Memory operations in programs are executed at virtual address and the memory footprints of programs reside in the secondary storage. Caches are used to eliminate most secondary storage accessing. The replacement routines for memory accessing in code virtualization mainly consist of following steps:

(i) locate the page of accessing the virtual address in caches;

(ii) if the cache is not present, load cache data from secondary storage page at virtual address;

(iii) issue the final read or write in the cache.

The code virtualization employs a progress in which virtual addresses are translated to physical addresses to caches, which is called address translation. The major part of the execution overhead of virtual memory takes place among address translations. The execution overhead of address translation mainly consists of the following parts:

(i) execution overhead of the wrapper snippets,

(ii) execution overhead of cache searching,

(iii) execution overhead of handling cache misses.

The wrapper snippet is mainly responsible for saving the current context and calling the cache searching routine. The implementation and overhead of the wrapper snippet are highly related to the specific instruction set used by the microcontroller.

The overhead for searching cache depends on how the caches are structured. The searching overhead would be minor if the number of caches is small and the caches are structured by cache sets [9], or if the caches can be located at compile time [10]. However, our virtual memory is designed to support complex data processing programs which have large memory footprints, so there must be enough RAM reserved for caches to achieve very low miss rate. We have chosen to organize caches in a fully associative way. It is known that using fully associative caches would achieve the minimal miss rate [11]. However, it is generally slower to search among fully associative caches because more caches need be travelled before hitting the target cache. Although the time of executing cache searching once is just tens to hundreds of MCU cycles, the address translation needs to be executed for every virtual memory access so the total overhead is large.

Overhead of handling cache misses is practically equal to the overhead of secondary storage accessing which is discussed in the next subsection.

2.2. Secondary Storage Accessing. Our work uses large parallel NAND flash with typically 64 sectors in one block as the secondary storage medium. Different flash techniques have been examined in [12] which gave an energy consumption comparison as shown in Table 1, proving that parallel NAND flash is very efficient in per-byte power consumption. To our knowledge, there is currently no work in the area of sensor networks which optimize secondary storage accessing for large NAND flashes. This is probability due to the fact that existing virtual memory mechanisms on sensor nodes are mostly designed for motes using small, low-power serial flashes which can be erased and written on a page basis.

We assume that the virtual address space is mapped to the NAND flash storage space linearly. This is reasonable because the size of NAND flash (gigabytes) is much larger than the required memory space of sensor network applications (megabytes). When accessing a variable at specified virtual address, the virtual address can be mapped to a NAND flash address directly by adding a constant offset. The data at the calculated NAND flash address is then loaded to a RAM cache and is finally served to accomplish the memory operation.

Table 1: Flash energy consumption read, write, and erase.

	Read	Write	Erase	Bulk erase (Page count)
Atmel NOR	0.26	4.3	2.36	n/a
Telos NOR	0.056	0.127	n/a	0.185 (256)
Hitachi MMC	0.06	0.575	0.47	0.0033 (16)
Toshiba 16 MB NAND	0.004	0.009	n/a	0.004 (32)
Micron 512 MB NAND	0.027	0.034	n/a	0.001 (64)

We however have observed some special features in virtual memory secondary storage access. First, the memory footprints of sensor node applications are much smaller than the NAND flash size. So it would be preferable to trade the flash space for reduced access time if possible, since energy is the major concern on sensor nodes. Second, the common knowledge that the access overhead of one NAND sector byte and the access overhead of the whole NAND sector are approximately equal does not hold for sensor nodes since MCUs on sensor nodes are relatively slow and the transmission time of command and data in NAND flash operations is significant and is basically proportional to the access size, although the underlying NAND flash sector reading or programming time is constant for different access sizes. Figure 1 shows the execution time of reading and writing a NAND flash (Model: K9F1G08X0A) on our image sensor node with different sizes for 16000 times. It can be observed that the execution time increases as the size of reading or writing grows and the command time (total write time minus write program time or total read time minus read transfer time) is proportional to the R/W size. The constant execution time for data transfer of reading and programming of writing is irrelevant to the R/W size. The data transfer time and program time takes a minor part of the total execution time which is particular for resource-constrained sensor nodes. One may also notice that the readings are not necessarily faster than writings. So, it is important for VM secondary storage access to reduce extra reading and writing due to their high overhead.

Besides the relative slow access speed, NAND flash has a special access style that pages must be erased before rewriting. NAND flashes are typically read and written on a page basis and erased on a block basis and one block contains multiple pages (typically 64). This requires an intermediate layer called flash translation layer (FTL) to be introduced to adapt NAND flashes to block devices and do wear leveling. State-of-the-art FTLs [13–15] are generally designed and optimized for file systems. Virtual memory secondary storage was not a concern of the designers of these FTLs. FTLs designed for file systems are usually concerned with the file system consistence upon power shutdown, which is irrelevant for

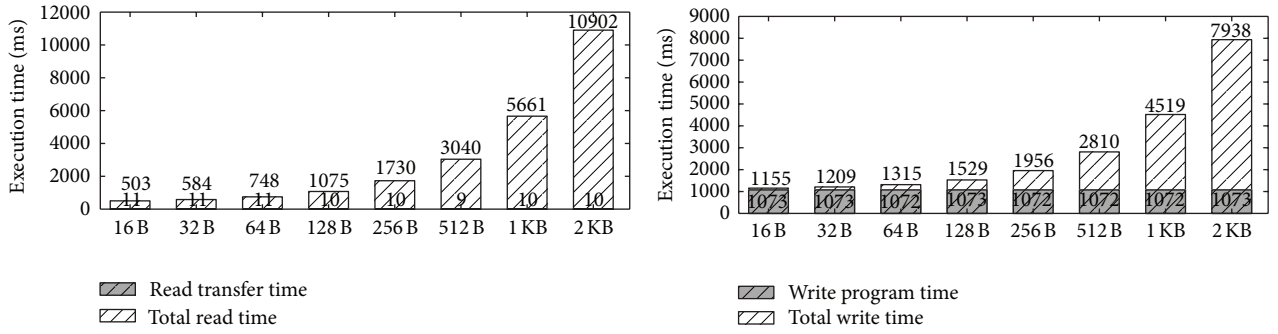
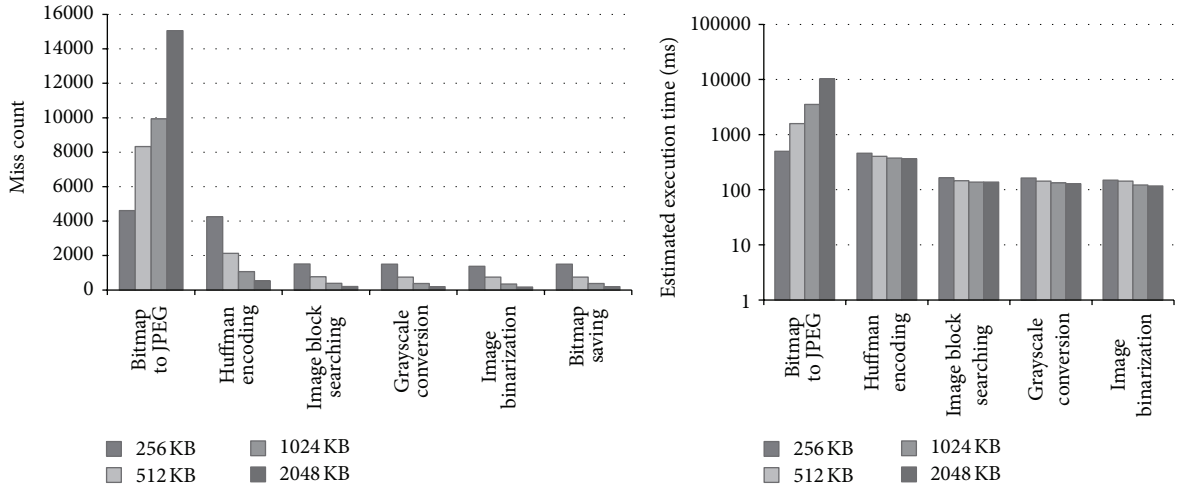


Figure 1: Execution time of reading and writing NAND flash with different sizes.



(a) Cache miss counts of programs

(b) Estimated read overhead for cache misses

Figure 2: Evaluation of cache misses of programs using LRU with cache RAM footprint being 32 KB.

VM secondary storage because data in virtual memory does not need to persist after reboot. NAND flash has an erase limit at about 10000 to 100000 times per block. So, it is critical for virtual memory to reduce block erasing to prolong the lifetime of sensor nodes.

The other issue is the mismatch between the cache size and the NAND sector size. It is apparent that if the two sizes are equal, caches can be read or written back by reading or writing just one sector. However, the best cache size is determined by the memory access pattern of the application and the available RAM space for caches [11]. Figure 2(a) shows the cache miss counts for some programs using LRU cache replacement algorithm with RAM footprint of caches being 32 kilobytes. As the cache size grows, the miss count increases for the JPEG to BITMAP program and decreases for other programs. This is because if the memory access pattern is relatively sequential or the program's working set is small enough, using larger caches would not cause cache misses, otherwise, larger cache size would lead to higher cache miss rate. Read overheads for the cache misses for these programs as shown in Figure 2(b). For the JPEG to BITMAP program, the estimated read overhead grows dramatically with increased cache size. However, for other programs, the estimated read overhead is more or less the same for different cache sizes. The estimated read overhead can be a rough approximation for the secondary storage access overhead. So, it is apparent that cache size smaller than the NAND sector size can have better performance than larger cache sizes; thus the NAND sector size is generally larger than the best cache size. This issue is specific to sensor networks since the page size of virtual memory on conventional computer systems (typically 4 KB) is large and thus this issue does not exist.

2.3. Fast Transparent Virtual Memory for Sensor Nodes. We have developed a fast transparent virtual memory mechanism named FaTVM [16] for our image sensor nodes. FaTVM is implemented on Cortex-M3 microcontroller platform. FaTVM aims to provide sensor network data processing programs with an efficient memory space which can be much larger than the physical memory.

To reduce the overhead of code virtualization, we allocate last-cache buffers (LCBs) for lvalue accessing in C code. LCBs are buffer variables which save the last matched caches for lvalues. When an LCB is hit, the cache searching overhead can

be eliminated. Our evaluation shows the probability of LCB-Hit is generally higher than 90%. So, using LCBs can achieve fast address translation for most virtual memory accessing. LCB scheme is based on C code transformation which does code virtualization on C code level rather than assembly instruction level. C code transformation has further reduced the execution overhead of the wrapper which was described in Section 5.

We have designed and implemented an FTL named Lavish-FTL which is best suited for sensor nodes with large NAND flashes. The main idea of Lavish-FTL is to trade NAND flash space for better performance. The main configuration of Lavish-FTL, which we called the "lavish degree," determines how much NAND flash space is going to be used for a fixed FTL size configuration. For example, if the lavish degree is 2, Lavish-FTL needs 2 megabytes of flash space to provide 1 megabyte of FTL address space. As the lavish degree grows, the execution overhead of Lavish-FTL can be reduced; however other FTLs' overhead cannot be reduced when more flash space is provided. The increased flash space overhead is not an issue for sensor nodes with large NAND flashes. Assuming that the virtual memory size is 8 MB and the lavish degree is 8, Lavish-FTL would need 64 MB flash space served as secondary storage, which is acceptable for large NAND flashes. Lavish-FTL tries to reduce the overhead of erasing and eliminate most extra readings or writings which might be inevitable in block management.

We introduced two schemes to address the mismatch between the cache size and the NAND sector size. One solution is that when writing one cache back, other conjoint dirty caches belonging to the same sector are searched in the cache list so that these caches can be written back by writing NAND sector only once. Evaluation shows that this scheme reveals good performance when there are plenty of caches. The other solution is that an adapting layer is used to reduce the sector size of the FTL. Evaluation shows that this scheme is more efficient than the former scheme if the number of caches is limited.

3. Related Work

3.1. Virtual Memory for Sensor Nodes. Virtual memory has been an important research subject in traditional operating system research for decades. It provides applications with isolated large flat address spaces which greatly simplify development of reliable programs and prevent programs from ruining each others' address spaces. However, traditional virtual memory systems are generally based on MMUs, so they are not usable in MMU-less embedded systems.

Softvm [17] implements software-managed address translation without TLBs however it is still designed for conventional computer systems.

MEMMU [18] proposed an automated compile-time and run-time memory expansion mechanism to increase the amount of usable memory in MMU-less embedded systems by using data compression; however available memory can only be increased by up to 50%.

There have already been several studies on virtual memory in sensor networks. Most previous work was trying to support large complex programs like TinyDB on sensor nodes. The virtual memory was generally assumed to be relatively small and node interactivity was the major consideration among application performance metrics.

t-kernel [9] provides software-based virtual memory called DVM by which user applications can have flat virtual memory spaces much larger than the physical memory space of the host node. Data frames are used to cache virtual memory accessing, which is the same as the caches in FaTVM. t-kernel searches among multiple data frames to find the hitting data frame. t-kernel searches among caches in a round-robin scheme in which all data frames are arranged in a data frame array and it starts searching from the last used data frame. This scheme however could lead to significant searching overhead if there are many data frames. Assembly virtualization is used by t-kernel to make programs access virtual memory. However, assembly virtualization introduces more overhead for context saving and restoring compared with C code virtualization. According to our earlier implementations, even more overhead will be introduced by assembly virtualization on our image sensor nodes due to the flexible addressing modes of the ARM instruction set.

Evaluation of t-kernel showed high efficiency in memory access performance. Accessing heap without swap takes only 15 cycles. We believe it is because that address translation of t-kernel DVM is simpler than that of FaTVM, which is partly due to the simpler address modes of sensor platform MCU. However, t-kernel DVM has a much higher miss rate than FaTVM. The current t-kernel implementation allocates 64 data frames in RAM as buffer for flash access, and the miss rate is about 10% for "slidingwin" application, tens of times of the miss rate of FaTVM. So, we argue that the caching scheme and the assembly virtualization method in t-kernel DVM are not applicable for sensor network programs with complex data processing because of the high overhead of secondary storage access. We did not evaluate t-kernel's performance in this paper because t-kernel is not ported to STM32 MCU.

ViMem [10] brings virtual memory support to TinyOS [19]. Developers add tags to nesC [20] source code to place variables in virtual memory. ViMem creates an efficient memory layout based on variable access traces obtained from simulation tools to reduce virtual memory overhead. The effect would be significant for traditional sensor node programs; however it would be useless when programs access large variables frequently, which is common in complex data processing. Furthermore, it is not possible to reference a variable in virtual memory using a normal pointer variable in ViMem. Pointer variables in virtual memory have to be tagged with the attribute "@vmptr." Since data elements in virtual memory are not necessarily contiguous, casting variables to types of a different size is not allowed, neither is pointer arithmetic. We argue that virtual memory transparency and flexible pointer operation are vital for porting or implementing complex data processing algorithms.

Recently, Enix [21] supplies software segmented virtual memory for code memory by code modification and uses Micro-SD cards as secondary storage. However Enix does not provide virtual memory for data segments because of high run-time overhead, which is the very issue FaTVM trying

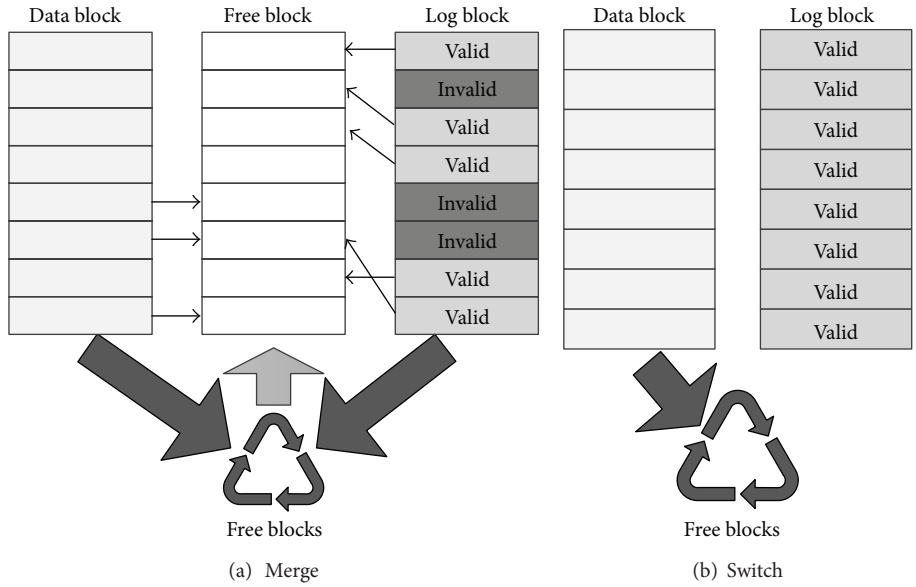


Figure 3: Block merge and switch in log based FTLs.

to address. Our earlier version of FaTVM also used Micro-SD cards as secondary storage; however Micro-SD cards are not as efficient as NAND flashes in energy consumption and access speed according to our evaluations of both implementations.

Our work differs from previous work largely in that we are trying to provide a large efficient virtual memory for mass data processing. The virtual memory mechanism should be able to increase the programs' available memory for tens of times or even hundreds of times. Most memory ranges accessed in data processing programs on sensor nodes are assumed to be in virtual memory. Directly applying previous work would lead to unacceptable overhead. Our work mainly focused on reducing the overhead of address translations and secondary storage accessing.

3.2. Flash Translation Layers. Due to the characteristics of NAND flash memory, flash translation layers are used to hide the inconvenient raw NAND flash interfaces and provide a block device interface. Taxonomy and design decisions of FTLs are not discussed in this paper due to paper size limitation.

Log-based FTLs are known to be efficient in sector writing. Log-based FTLs usually use one or more log blocks for the data block to buffer sector writings. Merge and switch are used to reclaim NAND blocks. Figure 3 shows the basic concepts of block merge and switch in the Log-FTL [13]. In log block merge, valid sectors are copied from the data block and log block to the free block, which brings about copy overhead of valid sectors. One free block is obtained after block merge with two erasings. When the log block contains all the sectors sequentially, block switch can be used which generates one free block with one erase and no copy of valid sectors. Log blocks in Fast FTL [14] can be used for any data block to increase the log utility. A dedicated sequential block is used to handle sequential writes efficiently with block switch. Superblock FTL [15] uses superblocks (a superblock consists of a set of adjacent logical blocks) to further exploit block-level spatial locality. Superblock FTL increases the chances of partial or switch merge operations instead of the expensive full merge operation.

4. FaTVM Overview and Architecture

We describe the overview of FaTVM in several aspects. Section 4.1 describes our developing hardware platform of FaTVM. Section 4.2 explains how source code is virtualized and compiled to final executive. Section 4.3 describes how virtual memory is accessed in programs.

4.1. Platform Description. We have been developing FaTVM on our image sensor nodes of an image sensor network [22]. Image sensor networks acquire image data by camera, and the sensor nodes are relatively more powerful than common motes such as MicaZ [23], TelosB [24], and so forth. The image sensor nodes are equipped with 32-bit ARM Cortex MCUs [4] from STMicroelectronics. The microcontroller has 72 MHz maximum frequency, 256 to 512 kilobytes of flash memory, and up to 64 kilobytes of SRAM. Although FaTVM is developed on our image sensor nodes with relative richer resource, it should be straightforward to port FaTVM to other platforms, since FaTVM is mostly implemented using C programming language and does not employ any advanced MCU features.

The image sensor nodes are equipped with large SLC NAND flashes for storing acquired images. The size of the NAND flash is 2 gigabytes. FaTVM uses a dedicated partition of NAND flash as the secondary storage.

STM32 MCU has a 4 GB memory map, in which memory address ranges of different sizes are allocated for code, SRAM, peripherals, external RAM, and so forth. FaTVM gives the programmer a view of contiguous memory space and a

dedicated range of address are used by application code to access virtual memory. We have specified a subrange of 1 GB external RAM addresses to be used by virtual memory, making it possible to determine address type to be virtual or physical by comparing the address with the boundaries of the dedicated address space.

4.2. Code Virtualization. FaTVM is developed without any assumption of the underlying operating system. Program code written in C programming language is transformed to access virtual memory. This progress is called code virtualization. Figure 4 shows how source files are virtualized and compiled to objects, which are relocated and linked to build the final executive. Programmers are responsible for specifying which source files to be virtualized thus it is possible to virtualize only a selected set of modules. Code virtualizer transforms source code written by programmers, who may not be aware of the virtual memory, to work with FaTVM. Object relocator moves variables in physical data sections to virtual data sections. It is still possible for programmers to manually specify some variables to be allocated in physical memory. This may bring significant performance improvement if variables accessed frequently in the program are allocated in RAM.

Assembly virtualization is commonly used in other software-based virtual memory mechanisms [9, 21]. Instructions of memory load or store are replaced by subtle assembly snippets to handle virtual memory accesses. It is a straightforward software method for implementing virtual memory. However, assembly virtualization has the following drawbacks.

(i) the overhead of assembly virtualization is high;

(ii) the replacements for memory load/store instructions are difficult to craft;

(iii) it is difficult to manually write complex assembly routine;

(iv) assembly virtualization may bring issues related to MCU instruction set specifics.

The current execution context (e.g., status register) needs to be saved before accessing VM and restored afterwards, otherwise the context will be ruined during accessing VM. Complex functionalities like cache searching and management are normally implemented in C, and invoking C routines from assembly brings about function invocation overhead. Cortex-M3 MCUs support many load/store instructions and flexible addressing, so it is a big challenge to write assembly VM accessing routines for all different memory load or store instructions. Furthermore, it is extremely difficult to implement sophisticated optimizations directly in assembly without function invocation.

Through FaTVM supports assembly virtualization, C code virtualization is preferred. C source files are transformed by virtualizer and compiled to objects without assembly transformations. C code virtualization enables more flexible code transformation and we have taken advantage of it to introduce last-cache buffers to fastly locate caches for virtual

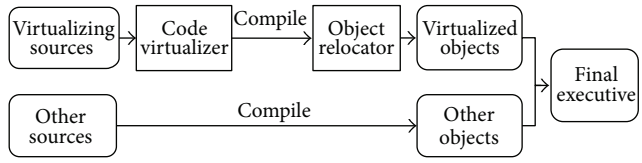


Figure 4: Program building process with FaTVM.

memory accessing. Section 5 describes details of the C code virtualization. Assembly virtualization is neglected in this paper.

4.3. Accessing Virtual Memory. FaTVM uses caches to boost virtual memory performance. A configurable number of block sized caches are used to store block data during multiple accesses. When accessing virtual memory, the page on NAND flash containing the accessing bytes is first read to cache. Then, the virtual address is converted to physical address to the corresponding bytes in cache. This conversion is called address translation. Finally the corresponding bytes in cache are read or written to accomplish the access request. Caches greatly reduce secondary storage accessing cost because FaTVM reads or writes secondary storage only when it failed finding a hitting cache. The number of cache blocks is limited by available free RAM size. It is apparent that allocating more cache blocks leads to less cache misses.

In general, application code accessing virtual memory is translated to reading or writing of the secondary storage. NAND flash must be read page by page, and they are cached in memory to accelerate multiple accessing. Virtual addresses are resolved to physical addresses located in cache blocks. The address translation progress is shown in Figure 5. Each cache block has a 4-byte field for storing per-cache control information. The structure of per-cache control field is also shown in Figure 5. The control field consists of an identify flag and a dirty flag. The identify flag is the visual address of the corresponding virtual memory block without lower bits of cache offset. The dirty bit indicates whether the cache block is written and needs to be written back to the secondary storage. According to our evaluations, address translation can be a major overhead of virtual memory in data-intensive programs. We have employed specialized optimization in C code virtualization which is able to reduce address translation overhead to a great extent.

The algorithms for cache searching and replacement have great influences on the performance of virtual memory. Rather than using set-associative caches [11], we use fully associative caches in which any cache can be mapped to a virtual memory data block. All cache blocks are linked in a universal single linked list (hereafter called cache queue). LRU is used for cache replacement algorithm. When resolving a virtual address, all cache blocks are traveled from the most recent used to the least recent used to find the matching cache. Searching in the cache queue is generally slower than searching in set-associative caches because set-associative caches can be implemented using efficient cache array data structure. Nevertheless, fully associative cache is known for

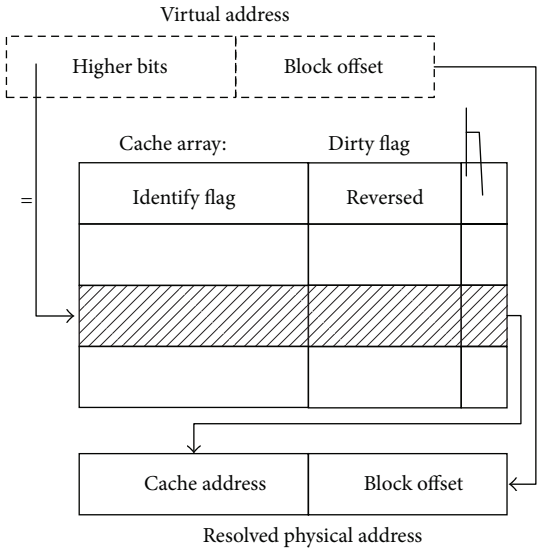


Figure 5: Address translation in FaTVM.

its lowest miss rates [11], and so it performs best in FaTVM due to high miss penalty.

CFLRU [25] and other cache replacement algorithms can also be employed easily to the current implementation; however they do not bring observable performance improvement for data processing programs according to our evaluation.

5. C Code Virtualization

We use CIL (C Intermediate Language) [26] to transform C code. Preprocessed C sources are passed to the C code virtualizer and then compiled by C compiler.

5.1. Image-Based Lvalue Virtualization. C code reads or writes memory in the form of lvalue, which means memory is accessed when any lvalue is read or written. C compilers normally translate lvalue access to assembly code that loads or stores at the start address of the specified lvalue; however in our virtual memory, memory addresses of lvalues in virtual memory are not located in physical memory, making direct load or store invalid. C code virtualization transforms C code so that lvalues in virtual memory are accessed using virtual memory routines. Generally, when reading an lvalue, the start address and size of the lvalue are passed to a predefined virtual memory routine and the corresponding bytes are read from virtual memory. Writing an lvalue is analogous to reading.

To facilitate C code virtualization, we create local variables in functions, which we call them local images. Local images are the copied values of the lvalues. Each different lvalue has its local image if the lvalue access needs to be virtualized and the type of the local image is the same as the type of the lvalue. Generally, there are two different forms of lvalue virtualization listed as belows:

(i) reading lvalue in an expression: read the lvalue to its local image and replace the lvalue with its local image in the expression;

```
vm_var = 10; vm_var is in virtual memory*/
another_var = global_var;
```

```
int local_image; // the local image of vm_var
local_image = 10;
//write back vm_var
vm_write_int(&vm_var, local_image);
//read lvalue to local image
local_image = vm_read_int(&vm_var);
another_var = local_image;
```

Figure 6: C lvalue Virtualization.

(ii) assigning lvalue in an instruction: replace the lvalue with its local image and write the local image back to the lvalue after assigning.

A simple example of C code translation is shown in Figure 6. Note that evaluating one expression may read two or more lvalues resulting in multiple virtual memory accesses.

Not all lvalues need virtualization. For those lvalues which can be inferred to be located in physical memory (e.g., local variable lvalues), virtualization is not applied. The current implementation is simply determined based on lvalue type. Global or static variables are determined to be allocated in virtual memory except those the programmer specified to be in physical memory. Local variables are determined to be in physical memory and accessed at native speed. lvalues involving pointer dereferencing (e.g., lvalue "*p") are always virtualized conservatively, because the target memory types of pointers are unknown.

5.2. Lvalue Synchronization Algorithms. Local images need to be synchronized to their lvalues when they are read or written so as not to destroy the original logic of program. Synchronization of an lvalue and its local image includes reading the value of lvalue to local image and writing the local image back to the lvalue. However, when an lvalue is used in a piece of code for multiple times, it is not necessary to synchronize the local image at all time. FaTVM uses data dependence analysis to determine if local images need to be updated or written back when they are used as substitutes of their corresponding lvalues. The algorithms of data dependence analysis for local image update and writing back are shown in Algorithms 1 and 2. A local image is read from the lvalue when it may have been unsynchronized in any execution path to the current statement. As in Algorithm 1, the local image is updated when there exists a statement which may be executed prior to the current statement and may desynchronizes the local image, and there exists an execution path from the desynchronizing statement and the current statement in which no statement synchronize the local image. The local image needs to be written back if the value of lvalue affects any subsequent execution. As in Algorithm 2, data dependence analysis is used to determine if the subsequent executions depend on the synchronization between the local image and the lvalue. These two algorithms are able to avoid superfluous local image synchronizations,

```
for each execution path to the current statement:
    may_not_initialized = true
    for each statement in execution path in reverse order:
        if the statement synchronizes the local image:
            may_not_initialized = false
            break
        if the statement desynchronizes the local image:
            return true
    if may_not_initialized:
        return true
return false
```

Algorithm 1: Algorithm to determine if reading local images from lvalues is necessary.

```
for each execution path from the current statement:
    reach_end = true
    for each statement in execution path:
        if statement depends on the synchronization of local image:
            return true
        if statement change the lvalue value:
            reach_end = false
            break
    if reach_end:
        return true
return false
```

Algorithm 2: Algorithm to determine if writing local images back to lvalues is necessary.

which become significant if the program references the same lvalue for many times.

There are four relations between code statements and lvalues referenced in the above two algorithms, listed as belows:

(i) the statement synchronizes the lvalue;

(ii) the statement desynchronizes the lvalue;

(iii) the statement depends on the synchronization of lvalue and its local image;

(iv) the statement changes the lvalue.

A statement synchronizes a local image if any one of the followings is true:

(i) the statement changes the lvalue;

(ii) the statement reads the lvalue.

For example, statement *p = val synchronizes both lvalues *p and val and its image since it changes the lvalue "*p" and reads the lvalue "val."

A statement desynchronizes a local image if any one of the followings is true:

(i) the statement changes some lvalue which is referenced in the lvalue;

(ii) the statement changes an lvalue referring to the same object with different but collided offset.

For example statement "p++" desynchronizes lvalue *p since it changes variable "p" which is referenced in lvalue "*p." Also the statement "union_.field1 = 100" desynchronizes lvalue "union_.field2" if "union_" is of union type with two fields named "field1" and "field2."

A statement depends on the synchronization of an lvalue and its local image if the effect of the statement will be incorrect when the local image is not synchronized to the lvalue. A statement changes an lvalue if the lvalue is assigned in the statement. For example, statement "p = 100" changes lvalue "p." These however involve data dependence analysis, which is difficult in the presence of pointers because pointers can cause subtle and complex data dependences. [27, 28]. We currently take a conservative strategy that assumes pointers may point to arbitrary positions in memory. Statements using pointer dereferencing lvalues depend on all lvalues and statements always depend on pointer dereferencing lvalues.

5.3. Last Cache Buffers. Data-intensive programs in sensor networks, like image processing, are generally featured by high locality in memory accessing, which has been made use of by the caches. Caching eliminates most secondary storage access; however the address translation overhead is high when VM is frequently accessed. Due to the rule of spatial locality, most successive address translations refer to the same cache block. A large portion of address translation overhead can be eliminated if redundant address translations can be suppressed.

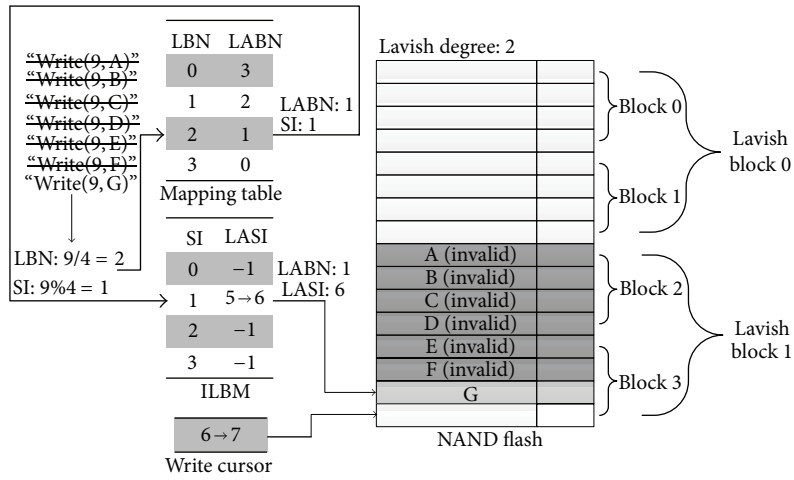


Figure 7: Address translation example of Lavish-FTL.

The flexibility of C code virtualization enables us to take advantage of this feature. Last-cache buffers are used to save the last matched cache of the resolving physical addresses while reading or writing lvalues. Last-cache buffers are variables of pointer type which are used to hold the last caches. Different last-cache buffers are used for different lvalues. Upon an lvalue access, the corresponding last-cache buffer is passed to the virtual address translation routine and the last matched cache is checked before the conventional address translation progress. If the cache block pointed to by the last-cache buffer matches the resolving virtual address, the address translation progress is highly boosted because the cache searching process is eliminated in this case.

We do not assign each lvalue an individual last-cache buffer. Different lvalues may be within the same cache block, so it would be reasonable and preferable to let two or more lvalues share one last-cache buffer. FaTVM assigns last-cache buffers to lvalues according to their "lvalue bases." The lvalue base of an lvalue can be derived using rules in Table 2. The main idea is that two lvalues of the same lvalue base should in high possibility be close to each other in memory.

Using last-cache buffers is able to eliminate most cache searching and thus reduce the cost of address translations which is responsible for a great portion of virtual memory overhead. More detailed evaluation of cost reduction benefiting from last-cache buffers is stated in Section 7.

Table 2: Lvalue base derivation rules.

Lvalue types	Derived lvalue base
variable	VarBase variable
variable.offset	VarBase variable
array (index)	VarBase array
*pointer	MemBase pointer
*(pointer + val)	MemBase pointer
*(pointer).offset	MemBase pointer

6. Secondary Storage Accessing

In this section, we describe the proposed Lavish-FTL and two adaptation schemes to address the aforementioned issues of using NAND flash as VM secondary storage.

6.1. Lavish-FTL. Lavish-FTL was specially designed for sensor nodes, to take advantage of the specific feature of NAND flash as secondary storage. The major motivation of developing Lavish-FTL is to trade NAND flash space for better access performance and less erasing. Lavish-FTL uses multiple NAND blocks to serve as one block, hereafter called a lavish block. The number of physical blocks in one lavish block is defined to be lavish degree. The larger the lavish degree is, the lower the NAND flash space utilization is. Logical block numbers (LBNs) are mapped to lavish block numbers (LABNs) rather than physical block numbers (PBNs). Figure 7 shows an example of Lavish-FTL sector address translation, where the sector with logical sector number (LSN) being 9 is written sequentially for seven times, with each write invalidating the previous write.

Sectors in a lavish block are written sequentially, from the first sector of the first physical block to the last sector of the last block. Each sector in the logical block is mapped to a logical sector of the lavish block. Since lavish blocks are written sequentially, the logical sectors in a lavish block are out of order. So, an intra-lavish-block mapping (ILBM) from logical sectors to physical sectors is required to search efficiently in lavish blocks. The ILBM contains the sector indexes in the lavish block (LASIs) for each logical sector index (SI). Note that there are normally more sectors in a lavish block than in a physical block. The ILBM information is needed for both reading and writing of sectors. The ILBM of a lavish block is saved in a sector of the lavish block itself, and a map index is maintained for each logical block to determine in which sector its latest ILBM is saved at. An example of an ILBM is also shown in Figure 7.

We use a configurable number of mapping caches to reduce the overhead of reading and writing of ILBMs. The caches are replaced using LRU algorithm. When a cache is

evicted from the LRU list, the cached ILBM is then written to the next unused sector of the corresponding lavish block, and the map index of the logical block is updated to index the last saved ILBM.

Same as the Log-FTL, we use log blocks for sector writing. That is, when a lavish block is full, a log lavish block (LLB) is allocated and subsequent writes are redirected to the log lavish block. The first lavish block is called the data lavish block (DLB). For simplicity, current implementation allows only one LLB for each DLB. The LLB increases the possibilities that the sectors in the DLB are invalidated when block erasing is required. Note that, since the ILBMs are maintained for logical blocks, there is only one ILBM for the DLB and LLB.

6.1.1. Reading Lavish Sector. Reading sector data in Lavish-FTL is simple. First, the logical sector number (LSN) is divided into LBN and the sector index. Second, the LBN is translated to LABN and the sector index is mapped to the sector in the lavish block. If the LABN is invalid or the sector was never written, the read buffer is filled with zeros. Otherwise, the corresponding sector in the lavish block is read to the read buffer.

Reading a specified sector in a lavish block is straightforward. If the sector index is smaller than the physical sector count in a lavish block (hereafter called lavish sector count), the sector is read from the DLB, and if the sector index is larger than the lavish sector count, it is read from the LLB. Figure 8 shows the lavish sector indexes for each physical sector in the DLB and its LLB.

6.1.2. Writing Lavish Sector. Writing sector data in Lavish-FTL is more complex than reading sector data. Same as the sector data reading, LSN is divided into LBN and the sector index. Second, the LBN is translated to LABN. If there is no lavish block for the writing logical block, new lavish block is allocated for the logical block and the mapping from LBN to LABN is constructed. The data is then written to the lavish block.

Lavish-FTL maintains a write cursor for each logical block as shown in Figure 7 which points to the next unused sector. The write cursor of a logical block is saved together with the ILBM. When there are unused sectors in the DLB, the data is simply written to the next unused sector. Otherwise if the DLB is full, an LLB is allocated for the DLB. If the LLB is also full, the DLB and the LLB are merged and the data is written after then. The merging details are described in Section 6.1.3.

6.1.3. Lavish Block Switching and Merging. There are two situations where NAND blocks are erased, called lavish switch and lavish merge.

When both the DLB and LLB are full in lavish block writing, a lavish switch progress is carried out so that there are free sectors in the DLB/LLB. Figure 9(a) shows an example of lavish switch. The lavish switch consists of the following:

(i) a free lavish block is allocated;

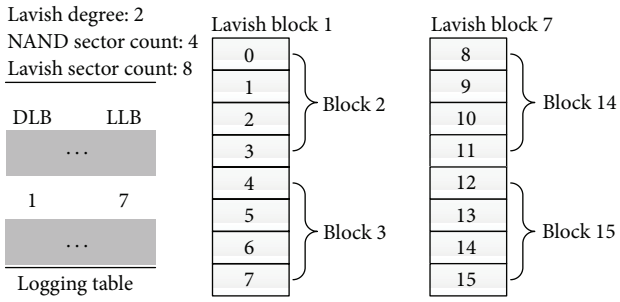


Figure 8: Lavish sector indexes of sectors in data block and log block.

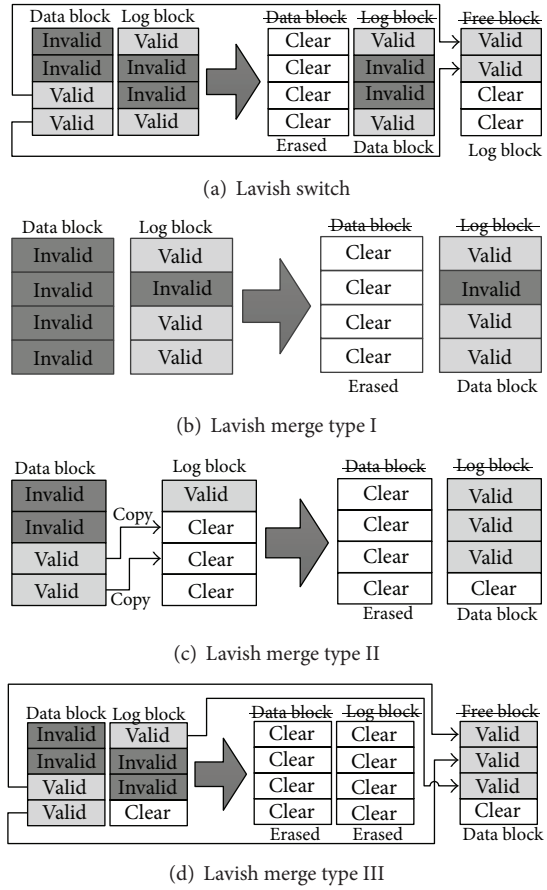


Figure 9: Demonstrations of lavish switch and lavish merge.

(ii) valid sectors in the DLB are copied to the new lavish block;

(iii) erase the DLB;

(iv) make the LLB as the DLB, and the new lavish block as the LLB.

The lavish switch is customized for writing a lavish block. Before lavish switch, the DLB always has valid sectors to copy, otherwise the DLB should have been erased by a lavish merge progress described later in this subsection. Lavish switch copies only the valid sectors in the DLB but not in LLB to

Table 3: Symbol definitions for performance analysis.

Symbol	Definition
LDEGREE	Lavish degree
SECTOR_COUNT	Count of sectors in a NAND block (typically 64)
EE	Erase efficiency of FTLs
CPBW	Number of sectors copied for every SECTOR_COUNT times of FTL writings

reduce sector copies. Since the LLB always has valid sectors before lavish switch, the new LLB always has free sectors after lavish switch even when lavish degree is 1, and thus the writing proceeds. Since the sectors in both DLB and LLB are out of order, the LLB can be used as DLB directly.

Lavish-FTL needs to make sure that there are always free lavish blocks. Lavish merge is used to reclaim free lavish blocks. There are currently three different types of lavish merge for different occasions as shown in Figures 9(b), 9(c), and 9(d).

Lavish merge type I is used when all sectors in the DLB are invalidated. In this occasion, the DLB is erased and the LLB is set as DLB. Valid sector counts of all lavish blocks are maintained in a byte array. Lavish merge type I is carried out whenever an DLB with no valid sector is found. It is similar to the "switch" operation of other FTLs.

When the number of free lavish blocks is low, lavish merge types II and III are used to reclaim free lavish blocks as shown in Figures 9(c) and 9(d). Lavish-FTL currently chooses the logical block among all logical blocks with both DLB and LLB which has the least valid sector count in the DLB. Lavish merge type II and type III differ in whether or not the LLB has enough free sectors to hold valid sectors in the DLB. The progress of lavish merge type II and type III can both be described as follows:

(i) if there are enough free sectors in the LLB to store the valid sectors in the DLB, set the old lavish block as the "writing" lavish block, otherwise a free lavish block is allocated to be the "writing" lavish block;

(ii) copy valid sectors from the DLB to the "writing" lavish block;

(iii) if the "writing" lavish block is not the LLB, copy valid sectors from the LLB to the "writing" lavish block;

(iv) erase the DLB;

(v) erase the LLB if the LLB is not the "writing" lavish block;

(vi) set the "writing" lavish block as the DLB.

6.2. Performance Analysis of Lavish-FTL. In this subsection, we analyze why Lavish-FTL can achieve better performance than other state-of-the-art FTLs. Some symbols are defined in Table 3 for clarification.

It has been stated by other researchers that the major overhead of NAND FTLs is due to the block erasing and valid sectors copying when merging multiple blocks [8, 13–15].

Erasing is generally the most expensive operation for NAND flashes. For block erasing, the best possible performance of FTLs is that each block erasing can get one free block. We define the erase efficiency to be the average number of free blocks to get per erasing. Apparently, the erase efficiency has an upper limit of 1.

Lavish-FTL has a lower bound for erase efficiency. The number of block erasing in lavish switch is the same as the lavish degree. When the lavish switch is complete, only the first NAND block of the new lavish block might be written with copied sectors and other NAND blocks were not touched. So, each lavish switch gets at least LDEGREE − 1 free NAND blocks. So the erase efficiency of Lavish-FTL has a lower limit of (LDEGREE − 1)/LDEGREE. Lavish merging has three different types. Lavish merge type I has the highest possible erase efficiency of 1. In lavish merge type II, where the valid sectors of the DLB is copied to the LLB, the number of erases is equal to the lavish degree and at most one NAND block in the LLB is used to store valid sectors from the DLB. This leads to an erase efficiency of (LDEGREE − 1)/LDEGREE, same as that of the lavish switch. In the third situation where a new lavish block is allocated to store all valid sectors from both the DLB and the LLB, the number of erasing is LDEGREE ∗ 2 and there are at most one NAND block in the LLB wasted and at most one NAND block in the final DLB used for storing valid sectors. So, the erase efficiency has a lower limit of (2 ∗ LDEGREE − 2)/(2 ∗ LDEGREE), same as the first situation and the lavish switch. So, Lavish-FTL has a lower limit of erase efficiency to be (LDEGREE − 1)/LDEGREE as shown in (1). It is easy to tell that, as the LAVISH DEGREE grows, Lavish-FTL becomes more efficient in block erasing. One has

$$\mathrm{EE}_{\text{Lavish-FTL}} \geq \frac{\text{LDEGREE} - 1}{\text{LDEGREE}}. \quad (1)$$

The other significant overhead of FTLs is due to copying of valid sectors. This is especially the case on sensor nodes since NAND data transmission for reading and writing is time consuming. Evaluation has shown that copying of valid sectors during block merging is the major overhead of FTL implementations on sensor nodes.

Lavish-FTL was designed to reduce the count of sector copying. We define the copy-per-block-write (CPBW) to be the number of sectors copied for every SECTOR_COUNT time of FTL writing. For Lavish-FTL, there are no more than SECTOR_COUNT times of copies during lavish switch or lavish merging. Since for each logical block, the lavish switch or lavish merging are carried out per LDEGREE − 1 times of block write at most and the number of sectors copied is SECTOR_COUNT at most. We can calculate the upper bound of CPBW to be (SECTOR_COUNT)/(LDEGREE − 1), as shown in

$$\mathrm{CPBW}_{\text{Lavish-FTL}} \leq \frac{\text{SECTOR\_COUNT}}{\text{LDEGREE} - 1}. \quad (2)$$

Another advantage of Lavish-FTL is that since a lavish block contains multiple NAND blocks, the possibility of sectors in DLB being invalidated by the time of lavish switch

or lavish merge is high, since LDEGREE ∗ SECTOR_COUNT number of sectors have been written to invalidate sectors in DLB. So the possibility of lavish merge type I being used is higher than that of the "switch" operation being used in other FTLs.

In our current implementation, lavish degree is set to 8 and SECTOR_COUNT is 64. So, the erase efficiency has a lower bound of 7/8 and CPBW has an upper bound of 64/7. In real world execution, Lavish-FTL performs better than the worst-case bounds. So, Lavish-FTL is extremely efficient and evaluation shows that Lavish-FTL is able to eliminate most extra overhead (erasing/copying) of reading and writing sectors.

6.3. Adapting Cache Size and Sector Size. As aforementioned, the best cache size is generally smaller than the sector size. So, the caches cannot be read or written directly. We have developed two schemes for addressing this issue, which are efficient in different situations, respectively.

The first scheme is called the multicaches scheme which enhances the LRU cache replacement algorithm by writing multiple caches to the lavish block at one time when writing back a dirty cache. Whenever the LRU algorithm chooses a cache to write it back, it searches in the cache queue to find dirty caches which belong to the same NAND sector with the writing back cache. When the caches cover the whole NAND sector, which means the number of caches is (sector − size)/(cache − size), these caches are written to the sector at one time. Otherwise, the missing data must be read from the sector before the caches being written to the sector. So, the possibility of caches within the same sector being found when writing back a cache is the key to the performance of this scheme. Evaluation shows that this scheme can be efficient when there is enough RAM space for caches. The evaluation details are left to Section 7.

The second scheme is called the sector-shrink scheme which reduces the sector size of FTL interface. An adapting layer is inserted between the FTL and the underlying NAND flash driver to reduce the sector size. In this way, the sector size of FTL is $1/2^n$ of the NAND flash sector size. For example, sector size of 2048 bytes can be adapted to 512 bytes. In this way, one physical NAND flash sector contains 4 adapted NAND sectors. Scheme II has its advantage and disadvantage compared to scheme I. The advantage is that since the FTL sector size is reduced, the overhead of extra reading in the scheme I can be reduced or eliminated (when the FTL sector size is adapted to be equal to the cache size). The disadvantage is that one NAND sector is written multiple times, cache by cache, which is not as efficient as writing the whole NAND sector. Evaluation shows that the sector-shrink scheme outperforms the multi-caches scheme when the number of caches is not abundant. The details are left to Section 7.

7. Evaluation

In this section, we evaluate the efficiency of FaTVM to verify its usability. We examined the elementary cost of address translation, cache performance, secondary storage accessing cost, and so forth to verify our design choices. We also evaluated the performance of the proposed FTL and sector size adaptation schemes. To evaluate the performance of real data-intensive applications, we have ported several well-known data processing algorithms to use FaTVM, and the execution results are shown and analyzed. Energy consumption of data processing programs using FaTVM is compared with other data handling options.

7.1. Evaluation Platform Description. We evaluate the secondary storage access performance on our image sensor nodes of an image sensor network. Image sensor networks acquire image data by camera, and the sensor nodes are relatively more powerful than common motes such as MicaZ, TelosB, and so forth. The image sensor nodes are equipped with 32-bit ARM Cortex MCUs from STMicroelectronics. The microcontroller has 72 MHz maximum frequency, 256 to 512 kilobytes of flash memory, and up to 64 kilobytes of SRAM. The sensor nodes are also equipped with an SLC NAND flash which is used by Lavish-FTL as the secondary storage.

7.2. Caching Evaluation. Performance metrics of caching include the following:

(i) cache searching overhead,

(ii) cache miss rate,

(iii) secondary storage accessing overhead.

Fully associative cache requires iterating in the cache queue one by one to find the matched cache. It is thus less efficient than set-associative cache. However, we do not evaluate the exact cost of cache searching because last-cache buffers in C code virtualization are able to eliminate most of cache searching overhead.

7.2.1. Cache Miss Rate. Cache miss rate is of critical importance to the performance of virtual memory since secondary access overhead is high. The miss rates for different cache configurations are shown in Figure 10. The result is acquired by emulating a memory access trace which is acquired by running JPEG to BITMAP program on FaTVM. Obviously miss rate raises as the memory footprint reduces. For each fixed RAM overhead, we examined different cache way settings to find the optimal setting. When the count of cache sets increases, the way of cache decreases. Count of cache set being 1 is equivalent to fully associative caches. The miss rates increase as the count of cache sets increases, and fully associative caches have achieved the least miss rates.

The figure shows that, with as much RAM as 32 kilobytes reserved for caches, the miss rates reached a low bound of about 0.28% for cache set counts less than or equal to 8. Set-associative caches can have smaller cache searching time than fully associative caches. So if abundant RAM is reserved for caches, the best cache configuration might not be fully associative caches, but set-associative caches. However, we did not examine cache configurations under different cache footprints. We adopted the fully associative

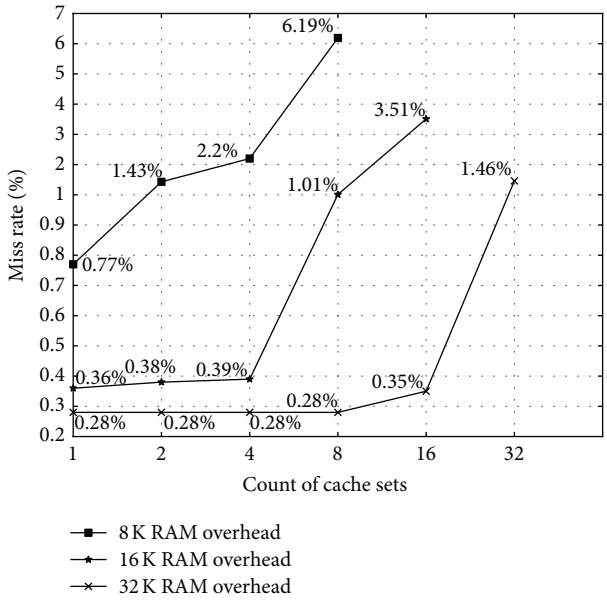


Figure 10: Cache miss rates of different configurations for JPEG to BITMAP.

cache for its simplicity and having lowest miss rates under all circumstances. Cache performance is also highly related to the executing program's work set and memory access pattern. Determining the best cache configurations for sensor node programs can be our future research work.

7.2.2. Basic Virtual Memory Access Overhead. The overhead of FaTVM accessing different memory regions is shown in Table 4. This is acquired by executing the memory operations for 2880000 times and recording the execution time. Note that the MCU frequency is 72 MHz. We also present the overhead of virtual memory access of t-kernel [9] and ViMem [10] for the ease of comparison. The data is copied directly from their published work.

Local variables in functions and static or global variables specified to reside in RAM are accessed at native speed. These variables are accessed frequently without incurring VM overhead. STM32F103ZE MCUs can execute native memory load/store instructions in 2 cycles. The measured 3 cycles include the instruction for loading the RAM address to registers. When last-cache buffers are not used, accessing virtual memory takes about 24 cycles in the best case, which means that the accessing virtual memory block is already cached and the cache is the most recent cache of the cache queue. When the accessing cache is not the most recent, multiple caches have to be travelled before finding the hitting cache, thus increasing the address translation overhead. If last-cache buffers are used, the virtual memory accessing overhead is reduced to 19 cycles when the last-cache buffer hits. Otherwise, the accessing overhead raises to be 33 cycles in the best case.

Recall that the overhead of "LCB-BestDisable" is the overhead of the best case not using last-cache buffers. If the accessing virtual memory block is not recently accessed, the extra overhead for travelling through the cache queue would incur more overhead. Although the overhead of "LCB-Hit" seems not to be much less than the overhead of "LCB-BestDisable" in best case, it eliminates much overhead of cache queue travelling.

Table 4: FaTVM basic operation overheads.

Access type	Loop	Execute time	Avg. cycles
Native	720000	30 ms	3
FaTVM (LCB-BestDisable)	720000	240 ms	24
FaTVM (LCB-Hit)	720000	190 ms	19
FaTVM (LCB-Miss)	2880000	1320 ms	33
t-kernel (AVR)	8388608	26.5 s	16
ViMem (AVR)	1	18.72 μs	17
FaTVM (SWAP)	1	3.3 ms	235714

LCB-BestDisable: best case when last-cache buffers are NOT used.
LCB-Hit: last-cache buffer hit when last-cache buffers are used.
LCB-Miss: last-cache buffer miss when last-cache buffers are used.
SWAP: swap due to cache missing.

Note that the same overhead of cache searching also exists in other virtual memory implementations. We believe this has not been emphasized in any other work because of two reasons shown below.

(i) The number of caches in the virtual memory implementations of this work is not as many as that in our implementation. Since we are expanding the available to megabytes, abundant caches must be provided to reduce the cache miss rate, making secondary storage accessing overhead tolerable.

(ii) This work tries to make the physical addresses of variables fixed, so that the address translation process can be simplified for these variables. That is, only one cache needs to be checked for validation. We argue that this technique does not work well for programs with megabytes of memory footprints. If the caches for some variables are fixed, accessing these variables would replace caches of other memory blocks, leading to higher miss rates. In-depth analysis or simulation is needed to determine the best cache assignments for variables, but it is out of the scope of this paper.

So, the last-cache buffers are used to keep address translation overhead small even when the cache is not the most recent. We evaluate the hit rates of last-cache buffers in real data processing programs, which can significantly affect the effectiveness of last-cache buffers. We also evaluate the effectiveness of last-cache buffers on real programs in later evaluation.

7.2.3. Last-Cache Buffer Hit Rate. Table 5 shows the hit rates of last-cache buffers on various programs. Two programs (i.e., FFT transformation and Red-block searching) reveal lower hit rates of more than 80%. Other programs reveal hit rates of larger than 90%. We believe that the program's hit rate of last-cache buffers is dependent on its access pattern of virtual memory. FFT transformation and Red-block searching operate on a small region of the data in virtual memory, which may frequently cross cache boundaries, invalidating

the last-cache buffer. For sequential accessing programs (i.e., BITMAP grayscaling), the hit rates can be larger than 99%.

7.3. Evaluating Flash Translation Layers. We run a selected set of typical programs using virtual memory and profile the performance of secondary storage access to test different flash translation layers. We have also implemented the Log-FTL and Superblock-FTL on our evaluation platform. 16 kilobytes of RAM are used for caches and the size of virtual memory is 8 megabytes. Table 6 shows the execution result of several programs with different FTLs. The erase count, copy count, and total execution time on secondary storage access are reported in the table. Note that neither adaptation schemes of the aforementioned mismatch issue is used in this evaluation.

Both Lavish-FTL and SuperBlock-FTL outperform Log-FTL to an extent. Although Lavish-FTL and SuperBlock-FTL have similar execution time for these programs, Lavish-FTL achieves the least erase count, guaranteeing the long-term operation of sensor nodes. The reduced ratio of erasing of Lavish-FTL compared with SuperBlock-FTL can be more than 30% for complex algorithms like JPEG decoding or Huffman compressing.

Table 8 shows the RAM footprints of the implementation of different FTLs. Lavish-FTL is most efficient in RAM footprint since run-time information is generally maintained for lavish blocks which are much fewer than the NAND blocks. The saved memory can be used for caches to further reduce the cache miss rate. Table 7 shows the execution result of the same programs using different FTLs where the whole available RAM is used for caches leaving sufficient memory for programs stacks. Apparently, Lavish-FTL is the most efficient in erasing and copying of valid sectors, and Lavish-FTL outperforms SuperBlock-FTL in execution time especially for complex programs like JPEG decoding.

7.4. Evaluating the Adaptation of Cache Size and Sector Size. We have proposed two schemes for the adaptation of cache size and sector size, because the best cache size is generally smaller than the sector size. Figure 11 shows the performance of secondary storage access with respect to different schemes running JPEG decoding. The same image used in Section 7.3 is used for evaluation. The cache size is set to 256 bytes, eighth of the NAND sector size, and Lavish-FTL is used for secondary storage access. When the RAM footprint of the caches is higher than 16 KB, multi-caches scheme outperforms sector-shrink scheme and when the RAM footprint of the caches is lower than 16 KB, the sector-shrink scheme outperforms multi-caches scheme, because multi-caches scheme would cause extra readings raising its overhead.

The secondary storage access takes about 1000 ms for both schemes when RAM footprint of the caches is 32 KB, which is about 16 times faster than the case when adaptation is not used and the cache size is set to the sector size (16382 ms as shown in Table 7).

7.5. Program Performance. We evaluate FaTVM performances using a set of typical image processing programs. The RAM footprint of caches is set to 32 kilobytes to reduce the miss rates. The size configuration of the virtual memory is 16 megabytes. The cache size is set to 512 bytes, which is an efficient setting for the programs. Last-cache buffers are used to reduce virtualization overhead. The total execution time of a program consists of the following three parts:

(i) algorithm time: execution time of the algorithm excluding memory accessing;

(ii) virtualization overhead: overhead introduced by C code virtualization;

Table 5: Last-cache buffer hit rates of programs.

Program	LCB-Hit rate
FFT transformation	80.63%
JPEG decoding	93.39%
Huffman compressing	91.99%
Red-block searching	83.33%
BITMAP grayscaling	99.87%

Table 6: FTL evaluations with 16 kB RAM for caches.

	Log-FTL	SuperBlock-FTL	Lavish-FTl
Erase count			
JPEG decoding	194	105	72
Huffman compressing	68	32	16
BITMAP grayscaling	389	295	264
BITMAP binarization	589	295	288
Copy count			
JPEG decoding	6208	761	172
Huffman compressing	2176	0	0
BITMAP grayscaling	6144	44	0
BITMAP binarization	18816	0	30
Execution time (ms)			
JPEG decoding	39344	32393	31425
Huffman compressing	10818	8076	8054
BITMAP grayscaling	65894	57910	57823
BITMAP binarization	82309	57995	58093

Table 7: FTL evaluations with all available RAM for caches.

	Log-FTL	SuperBlock-FTL	Lavish-Ftl
Cache RAM footprint	26 KB	17 KB	32 KB
Erase count			
JPEG decoding	194	105	32
Huffman compressing	34	15	0
BITMAP grayscaling	193	146	128
BITMAP binarization	294	149	136
Copy count			
JPEG decoding	6208	761	97
Huffman compressing	1088	0	0
BITMAP grayscaling	3072	44	0
BITMAP binarization	9408	0	0
Execution time			
JPEG decoding	35789	32400	16382
Huffman compressing	5456	4038	3990
BITMAP grayscaling	32909	28964	28515
BITMAP binarization	41154	28913	28631

Table 8: FTL RAM footprints.

Log-FTL	SuperBlock-FTL	Lavish-FTL
13532 bytes	23376 bytes	6276 bytes

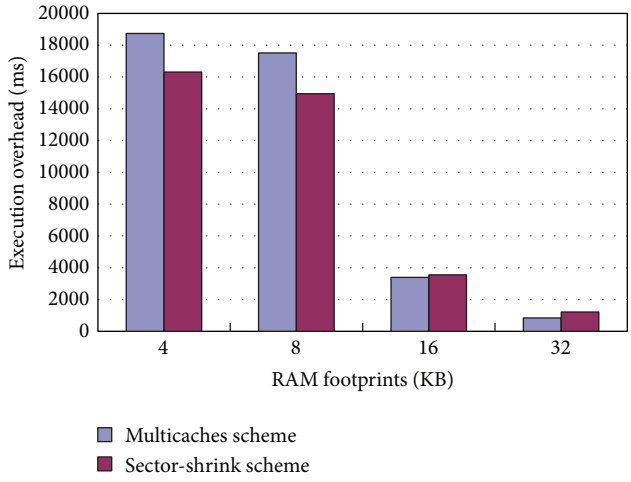


Figure 11: Execution overhead of the two adaptation schemes.

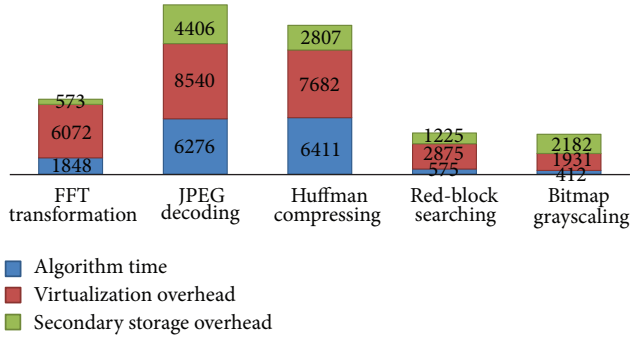


Figure 12: Execution times of typical programs (unit: ms).

Table 9: Operation conditions of the evaluation platform and Intel Mote 2.

	Intel Mote 2	Evaluation platform
MCU frequency	104 MHz	72 MHz
Sleep mode current	387 μA	25 μA
Active mode current	53 mA	37 mA
NAND operating current	N/A	15 mA
NAND stand-by current	N/A	10 μA
Battery voltage	5.5 V	3.3 V

(iii) secondary storage overhead: time spent on reading or writing NAND flash.

We measure the program's execution time using system timer interrupts. A flag is set before invoking virtual memory accessing and cleared after then. The system timer interrupts are fired per millisecond. The interrupt handler checks the flag and increases the corresponding counter by one. In this way, we can measure the execution times of algorithms and virtual memory. Figure 12 shows the execution times of programs.

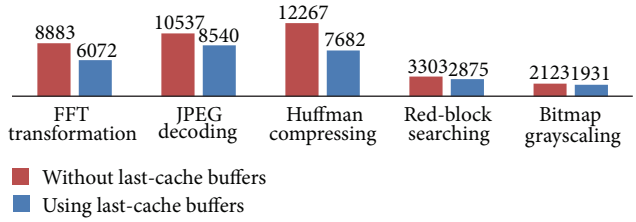


Figure 13: Effectiveness of last-cache buffers (unit: ms).

Table 10: Image sizes in JPEG format.

Image name	Image Size
big.jpg	43,584 bytes
small.jpg	27,504 bytes
tiny.jpg	3,080 bytes

Apparently, the algorithm time is different for each algorithm. JPEG decoding is the most complex among the evaluated algorithms and the algorithm of BITMAP grayscaling is the simplest. The virtualization overhead is relatively large compared with the algorithm time. For simple algorithms, the virtualization overhead is responsible for the major execution time of the programs. For complex algorithms, the virtualization overhead is smaller compared with the algorithm time and comparable to algorithm time for complex algorithms (i.e., JPEG decoding, Huffman compressing). The secondary storage overhead is much smaller than the virtualization overhead. For more complex algorithms including JPEG decoding, Huffman compressing, and FFT transformation, the secondary storage overhead is also much smaller than the algorithm time.

To verify the effectiveness of last-cache buffers, virtualization overhead of these programs without using last-cache buffers in virtual memory is measured. The result is presented in Figure 13. Last-cache buffers are able to reduce up to 40% of virtualization overhead.

7.6. Energy Consumption. It is shown in Figure 12 that the overhead introduced by FaTVM is significant compared with the algorithm time despite many optimizations we have adopted. To verify the availability of FaTVM, we compare the energy consumption of image processing programs using FaTVM with other options that process image or transmit image to powerful gateway listed below:

(1) run image processing algorithms on sensor nodes with sufficient RAM;

(2) send image to powerful gateway without processing.

We calculate and compare the energy consumption for these programs in our evaluation platform with the Intel Mote 2 platform which is equipped with 32 megabytes of SDRAM. The operation conditions of both Intel Mote 2 and our evaluation platform using STM32F103ZE is shown in Table 9. The calculation is based on following assumptions or conditions:

(i) the Intel Mote 2 runs 104/72 times faster than the evaluation platform for all algorithms;

Table 11: Image processing time of algorithms.

Image name	JPEG decoding (NAND)	Grayscale calculation (NAND)
big.jpg	11803 ms (NAND 3096 ms)	1462 ms (NAND 745 ms)
small.jpg	1118 ms (NAND 140 ms)	154 ms (NAND 76 ms)
tiny.jpg	60 ms (NAND 1 ms)	8 ms (NAND 0 ms)

(ii) energy consumption of peripheral devices is not considered;

(iii) energy consumption of other tasks (e.g., communication, storage) of sensor nodes is assumed to be irrelevant;

(iv) sensor nodes enter deep sleep mode most of the time and the extra energy consumption of NAND flash power up/down is neglected;

(v) the data processing algorithm is executed only once in the one duty cycle of the sensor node.

Then, the energy consumption (EC) of data processing and deep sleep mode of the sensor nodes regarding duty cycle (T) can be calculated using (3) and (4). $EC_{IntelMote2}$ is the energy consumption of Intel Mote 2 and EC_{FaTVM} is the energy consumption of the evaluation platform with FaTVM. AT is the algorithm time. SST is the NAND flash operation time which is equal to the secondary storage overhead of the virtual memory. TT is the total execution time using virtual memory. One has

$$\begin{aligned} EC_{IntelMote2} = {} & AT * \frac{72}{104} * 5.5\,V * 53\,mA \\ & + (T - AT) * 5.5\,V * 387\,\mu A, \end{aligned} \tag{3}$$

$$\begin{aligned} EC_{FaTVM} = {} & TT * 3.3\,V * 37\,mA \\ & + SST * 3.3\,V * 15\,mA \\ & + (T - TT) * 3.3\,V * 35\,\mu A. \end{aligned} \tag{4}$$

The duty-cycle energy consumption of previously evaluated data processing algorithms is shown in Figure 14. When the duty cycle is small enough, the Intel Mote 2 platform is more energy efficient than our evaluation platform because the execution time of data processing is significant compared with the duty cycle. However, due to higher sleep-mode current, the energy consumption of Intel Mote 2 increases much faster than that of our evaluation platform as the length of duty cycle grows. Most energy is consumed in sleep mode when the duty cycle is large enough in the Intel Mote 2 platform. The lengths of duty cycles with which two platforms have the same energy consumption are also calculated and shown in the figure, which are approximately between 5 and 20 minutes. This can actually be a threshold of the execution frequency of data processing on sensor nodes to determine if virtual memory should be employed. So, if sensor nodes have short duty cycles and data processing is executed frequently, high performance platforms like Intel Mote 2 with large SDRAM should be preferred. On the other case where data processing is infrequent, using STM32F103ZE MCU with FaTVM is about 3 times more energy efficient than using Intel Mote 2 for 1 hour duty cycles. We are developing an image sensor network to enable long-term monitoring of the surface changing of mural paintings, in which sensor nodes take photos of the mural painting per day. In scenarios like this, using FaTVM to support data processing is dramatically more energy-efficient than using Intel Mote 2 platform for sensor nodes. So, it is verified that the proposed virtual memory mechanism is helpful for constructing long-term sensor networks with long duty cycles and mass data processing.

We further compare the energy consumption of FaTVM with sensor nodes that collect and forward the sensed data to powerful gateways. Assuming an occasion in which a sensor network is constructed and sensor nodes collect images of targets, only the gray scale of the targets is interested and the image format given by cameras on sensor nodes is JPEG. There would be basically two approaches on your evaluation platform to do this task listed below:

(1) decode images in JPEG to BITMAP and calculate the gray scale of the images using FaTVM; then, send scalar gray scale values to gateways;

(2) send images in JPEG to gateways and let gateways do gray scale calculations.

Energy consumption of approach 1 contains energy consumption of image processing using FaTVM and data transmission. Since the data size of scalar gray scale values is neglectable compared with the image size in JPEG, energy consumption of transmitting scalar gray scale values is ignored. The energy consumption of FaTVM can be calculated using aforementioned equation (4) without counting in sleep mode energy consumption. Calculating energy consumption of forwarding sensed data to gateway is more troublesome in approach 2. We use data from others' work directly. It has been mentioned in [12] that the energy consumption (μJ/byte) for CC2420 [29] radio Tx/Rx is 1.8/2.1. However, it is only a theoretical minimum with 250 kbps data rate. According to [30, 31], practical observed data rate is only 40 kbps. Thus energy consumption of Tx/Rx is raised to 11.3 and 13.1 (μJ/byte). This data can be used to calculate the energy consumption of single-hop data transmission. In real sensor networks, reliable multihop transmission, network collision, and so forth can impose significant overhead to network transmission. So the energy consumption of network transmission in real sensor networks is much higher than that of single-hop data transmission.

We consider three different sizes for JPEG image, as shown in Table 10. The image processing time is shown in Table 11. So, the energy consumption for processing or transmitting each image can be calculated. The result is shown

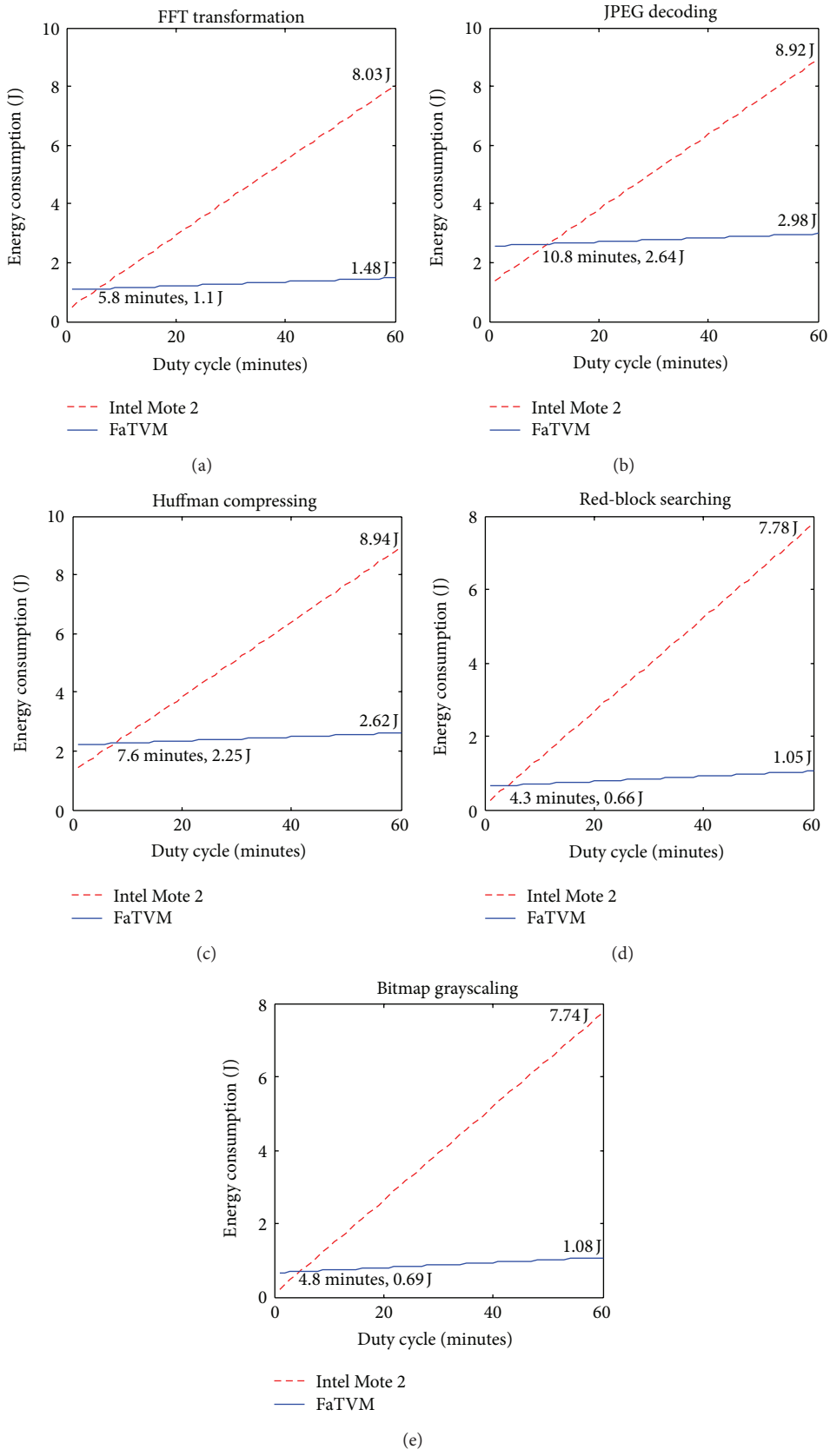


Figure 14: Program energy consumptions of one duty cycle.

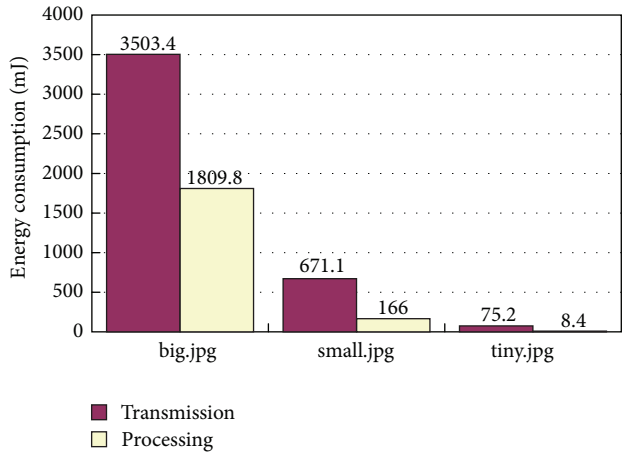


Figure 15: Energy consumption of image transmission or processing.

in Figure 15. Despite the JPEG decoding, data processing is still much more energy efficient than network transmission. Energy consumption of image network transmission is 2 times that of image processing for big images and more than 4 times that of image processing for smaller images. Recall that the calculated energy consumption of network transmission is just for one hop. Energy consumption of image transmission can be multiples of that in real-world sensor networks.

8. Conclusion and Future Work

We have designed and implemented a virtual memory named FaTVM for data-intensive applications in sensor networks, making it possible for sensor nodes to carry out complex computation with heavy memory footprint without using energy-hungry MCUs with large RAM. We have been focusing on reducing virtual memory overheads in different aspects including cache management, address translation, secondary storage accessing, and so forth, to achieve acceptable memory performance for sensor network applications. We use C code virtualization to transform C source code to work with virtual memory. Last-cache buffers are used to accelerate address translations which contributes to a great portion of virtual memory overhead. FaTVM uses NAND flashes as secondary storage. Lavish-FTL is proposed to reduce NAND flash accessing time and block erasing, so as to prolong NAND flash lifetime. Two adaptation schemes were introduced to write caches back to NAND sectors efficiently. Our evaluation shows that the virtual memory overhead is comparable to the algorithm time for typical data processing algorithms on image sensor nodes. The secondary storage overhead has been reduced to be the smaller part of the virtual memory overhead. We compared the energy consumption of the evaluation platform using FaTVM with the Intel Mote 2 platform under reasonable assumptions. The virtual memory solution is proven to be much more energy-efficient for mass data processing programs on sensor nodes with long duty cycles than high performance platforms with large SDRAM. Energy consumption of data processing programs using FaTVM is also proven to be much less than that of large data transmission.

Despite the optimizations we have adopted to reduce virtual memory overhead on various aspects, the overhead introduced by code virtualization is still large compared with the algorithm time. Reducing the overhead introduced by code virtualization without increasing cache misses can be our future research direction. Fat pointers are used to further reduce virtualization overhead by storing cache information for memory accessing. The memory access pattern of sensor nodes is explored to improve cache performance. We have also noticed that it is difficult to determine the best VM configuration for performance and energy efficiency. Finding an automatic solution to find the optimum VM configuration can further improve the utility of FaTVM.

Acknowledgments

This work was supported by the "Advanced Sensor Network Platform Research" Project of Zhejiang University, China (no. 2010X88X002-11), and the National High Technology Research and Development Program of China (Grant no. 2012AA101701).

References

[1] R. M. Kling, "Intel motes: advanced sensor network platforms and applications," in *Proceedings of the IEEE/MTT-S International Microwave Symposium*, pp. 365–368, June 2005.

[2] P. J. Denning, "Virtual memory," *ACM Computing Surveys*, vol. 2, pp. 153–189, 1970.

[3] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "Tinydb: an acquisitional query processing system for sensor networks," *ACM Transactions on Database Systems*, vol. 30, no. 1, pp. 122–173, 2005.

[4] STMicroelectronics, "Stm32f103ze datasheet," 2011, http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/CD00191185.pdf.

[5] "Atmega128 datasheet," http://www.atmel.com/.

[6] "Texas instruments: Msp430 datasheet," http://focus.ti.com/lit/ds/symlink/msp430f1101a.pdf.

[7] A. Ltd, "Cortex-m3 technical reference manual," 2010, http://infocenter.arm.com/help/topic/com.arm.doc.ddi0337i/DDI0337I_cortexm3_r2p1_trm.pdf.

[8] T.-S. Chung, D.-J. Park, S. Park, D.-H. Lee, S.-W. Lee, and H.-J. Song, "A survey of flash translation layer," *Journal of Systems Architecture*, vol. 55, pp. 332–343.

[9] L. Gu and J. A. Stankovic, "*t-kernel*: providing reliable OS support to wireless sensor networks," in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, pp. 1–14, 2006.

[10] A. Lachenmann, P. J. Marrón, M. Gauger, D. Minder, O. Saukh, and K. Rothermel, "Removing the memory limitations of sensor networks with flash-based virtual memory," *SIGOPS Operating Systems Review*, vol. 41, pp. 131–144, 2007.

[11] J. Hennessy, D. Patterson, and D. Goldberg, *Computer Architecture: A Quantitative Approach*, Morgan Kaufmann, 2003.

[12] G. Mathur, P. Desnoyers, D. Ganesan, and P. Shenoy, "Ultra-low power data storage for sensor networks," in *Proceedings of*

the 5th International Conference on Information Processing in Sensor Networks (IPSN '06), pp. 374–381, ACM, New York, NY, USA, 2006.

[13] C-.H. Wu, T-.W. Kuo, and C-.L. Yang, "A space-efficient caching mechanism for flash-memory address translation," in *Proceedings of the 9th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC '06)*, pp. 64–71, IEEE Computer Society, Washington, DC, USA, 2006.

[14] S. Lee, W. Choi, and D. Park, "Fast: an efficient flash translation layer for flash memory," *Emerging Directions in Embedded and Ubiquitous Computing*, pp. 879–887, 2006.

[15] J-.U. Kang, H. Jo, J-.S. Kim, and J. Lee, "A superblock-based flash translation layer for nand flash memory," in *Proceedings of the 6th ACM & IEEE International Conference on Embedded Software (EMSOFT '06)*, pp. 161–170, ACM, New York, NY, USA2006.

[16] N. Lin, Y. Dong, and D. Lu, "Fast transparent virtual memory for complex data processing in sensor networks," *SENSORNETS*, pp. 24–34, 2012.

[17] B. Jacob and T. Mudge, "Uniprocessor virtual memory without tlbs," *IEEE Transactions on Computers*, vol. 50, pp. 482–499, 2001.

[18] L. S. Bai, L. Yang, and R. P. Dick, "Memmu: Memory expansion for mmu-less embedded systems," *ACM Transactions on Embedded Computing Systems*, vol. 8, no. 3, article 23.

[19] T. T. 2.x Working Group, "Tinyos 2.0," in *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems (SenSys '05)*, p. 320, ACM, New York, NY, USA, 2005.

[20] D. Gay, M. Welsh, P. Levis, E. Brewer, R. Von Behren, and D. Culler, "The nesC language: a holistic approach to networked embedded systems," in *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '03)*, pp. 1–11, New York, NY, USA, June 2003.

[21] Y. T. Chen, T. C. Chien, and P. H. Chou, "Enix: A lightweight dynamic operating system for tightly constrained wireless sensor platforms," in *Proceedings of the 8th ACM International Conference on Embedded Networked Sensor Systems (SenSys '10)*, pp. 183–196, New York, NY, USA, November 2010.

[22] S. Soro and W. Heinzelman, "A survey of visual sensor networks," *Advances in Multimedia*, vol. 2009, Article ID 640386, 21 pages, 2009.

[23] Crossbow Technology, "Micaz specs," 2009, http://bullseye.xbow.com:81/Products/Product_pdf_files/Wireless_pdf/MICAz_.Datasheet.pdf.

[24] "Telosb specs," 2009, http://www.willow.co.uk/TelosB_Datasheet.pdf.

[25] S.-Y. Park, D. Jung, J.-U. Kang, J.-S. Kim, and J. Lee, "Cflru: a replacement algorithm for flash memory," in *Proceedings of the 2006 International Conference on Compilers, Architecture and Synthesis for Embedded Systems (CASES '06)*, pp. 234–241, ACM, New York, NY, USA, 2006.

[26] G. Necula, S. McPeak, S. Rahul, and W. Weimer, "Cil: intermediate language and tools for analysis and transformation of c programs," in *Compiler Construction*, R. Horspool, Ed., vol. 2304 of *Lecture Notes in Computer Science*, pp. 209–265, Springer, Berlin, Germany, 2002.

[27] D. C. Atkinson and W. G. Griswold, "Effective whole-program analysis in the presence of pointers," in *Proceedings of the 6th ACM SIGSOFT International Symposium on Foundations of Software Engineering (SIGSOFT '98/FSE-6)*, pp. 46–55, ACM, New York, NY, USA, 1998.

[28] S. Horwitz, P. Pfeiffer, and T. Reps, "Dependence analysis for pointer variables," in *Proceedings of the ACM SIGPLAN 1989 Conference on Programming Language Design and Implementation (PLDI '89)*, pp. 28–40, ACM, New York, NY, USA, 1989.

[29] T. Instruments, "Cc2420 datasheet," 2007.

[30] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 95–107, ACM, New York, NY, USA, 2004.

[31] K. Kumar and P. Kumar, "Tmote Implementation of bmac and smac Protocols," 2011.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 791516, 8 pages
http://dx.doi.org/10.1155/2013/791516

Research Article

An Energy-Efficient Broadcast MAC Protocol for Hybrid Vehicular Networks

DaeHun Yoo and WoongChul Choi

Department of Computer Science, Kwangwoon University, Seoul 139-701, Republic of Korea

Correspondence should be addressed to WoongChul Choi; wchoi@kw.ac.kr

Received 9 November 2012; Accepted 18 January 2013

Academic Editor: Jiman Hong

In vehicular networks, sparse roadside vehicular communication (RVC) systems on highways are difficult to provide full coverage of existing roadways because the distance between the roadside units (RSUs) is farther than the RSU's transmission range. In sparse RVC, collision-free communication is required because the time available for communication is very short. In addition, the collision-free communication has an energy-efficient feature. Therefore, we propose an energy-efficient broadcast MAC protocol in order to expand the service coverage in RVC systems and collision-free communication. The proposed protocol performs a hybrid vehicular communication (HVC). It has a rebroadcast mechanism using a vehicle's velocity, distance, and angle from nearby vehicle for collision-free communication. Then, we show our protocol's performance evaluation using ns-2.

1. Introduction

Vehicular communication is an integral part of intelligent transport systems (ITSs). GPS/DMB device has been already installed in many vehicles, and communication device will be also installed in most of vehicles in near future. The vehicular communication systems are divided into intervehicular communication (IVC) systems and roadside vehicular communication (RVC) systems [1]. IVC systems are completely infrastructure-free so that only onboard units (OBUs) are needed. In RVC systems, communication takes place between roadside units (RSUs) and OBUs. Hybrid-vehicular communication (HVC) systems are the hybrids of IVC and RVC systems. The studies on IVC systems have been conducted relatively more than the studies on RVC systems [2, 3].

This paper covers RVC systems with focus on information downloading service that provides information on road conditions or traffic. The information downloading service means seamless services such as multimedia streaming service [4]. RVC systems are divided to ubiquitous RVC (URVC) systems and sparse RVC (SRVC) systems. For URVC systems, RSUs are installed on all roads to ensure that communication with RVC systems, while a vehicle is moving, is not disconnected. Unfortunately, a URVC system may require a considerable investment to provide full coverage of existing roadways. On the contrary, for SRVC systems, RSUs are installed at a certain interval of distance, which is practical since it requires less investment cost than URVC systems. A weak point of SRVC system is that downloading service is unavailable if a vehicle is out of the RSU's transmission range because the distance between RSUs is large. Most of the studies on RVC systems had put focus on how to improve performance when a vehicle arrive within the RSU's transmission range [5–7]. However, if the RSU's transmission range is 1 km and a vehicle's speed is 100 km/h, the time available for communication is 72 seconds at the maximum, which is very short. Therefore, in order to provide an energy-efficient downloading service, the following study is necessary. First, the method expanding the service coverage is needed in order to extend the time available for communication. Second, collision-free communication is required in order to avoid the unnecessary delay occurred by the retransmission of the frame collisions. Minimizing the number of retransmissions means the energy-efficient features. From these motivations, this paper proposes an energy-efficient multihop relay broadcast MAC (MRB-MAC) protocol.

The remainder of this paper is organized as follows. In Section 2, a description of related research studies is provided. An application scenario is presented in Section 3. And an overview of our protocol is presented in Section 4.

The simulation validation is presented in Section 5. Finally, Section 6 concludes the paper.

2. Related Work

The routing protocols using the various metrics already have been studied for the multihop communication in the wireless networks [8, 9]. However, this paper focuses on the MAC layer protocol in order to reduce the routing overhead of the frequent topology changes according to a vehicle's movement characteristics.

Sloted-1 [10], which is a beaconless approach method, classifies the broadcast range locally and designates it as one "slot." And it allocates a shorter waiting time to the OBU in the farthest region based on the OBU that transmits data in order to avoid collision. The DDB [11] utilizes the distance of the OBU that an OBU will transmit after receiving data in order to calculate the cover additional area (AC) and the delay time for rebroadcasting. Unlike the case with the aforementioned study, the BCF [12] is a beacon-based approach method. This protocol periodically broadcasts the beacon message that includes the location, direction, and speed. And when a car accident happens, it selects a forwarder using the information of the beacon message.

RVC systems have the higher bit error rate (BER) than IVC systems because of the damage of wireless channel fading caused by the difference in high speed between an RSU and an OBU. Since communication in case of movement in the same direction is considered in IVC systems, the difference in speed is not greater than that of RVC systems. As a way to solve the problem, protocols that use cooperative communication (CC) were proposed. VC-MAC [5] proposed the mechanism to select relay node in order to avoid collision with other OBUs in consideration of the signal to noise (SNR) of the OBUs that received frame normally from the RSUs. ADC-MAC [6] uses CoopTable of CoopMAC [13] to select helper node and provides support for direct transmission (DT), cooperative relay (CR) transmission, and two-hop relay (TR) transmission based on the RTS-HRTS-CTS-HCTS-HACK mechanism. On the assumption that the location of the RSU is known to all vehicles, PVR [7], which does not use CC, proposed the method to select proxy node based on the information on neighbor nodes until it reached the RSU and to collect the information on nearby nodes in order to transmit data to the RSU. The authors of [14] presented the cluster-based mechanism employed for multimedia transmissions and the cluster head selection algorithm in HVC systems.

The problems found by the related works can be summarized as follows.

(i) There are insufficient studies on SRVC systems. In case of downloading service in SRVC systems, the RSU does not have enough time to provide service to the OBU because of having the limited transmission range.

(ii) Most of the studies did rarely take two-way street into consideration. In the two-way street, since the relative speed of a vehicle in the opposite direction is very high, there is a higher possibility of collision if the network topology changes more quickly in the contention mode.

(iii) The beacon-based mechanism to know the status of neighbor nodes such as BCF, VC-MAC, and ADC-MAC and the cluster-based mechanism causes wasting of bandwidth and frequent collisions.

3. Application Scenario

Before explaining the MAC protocol designed in this paper, this section explains an application scenario to which our protocol is applied. The application scenario presented in this paper is similar to information downloading scenario presented in VC-MAC [5]. The details, similar to those of VC-MAC, are as follows.

(i) In our scenario, we only consider the downlink, which means that an RSU sends data to an OBU.

(ii) In vehicular networks, every vehicle is equipped with a wireless device.

Their differences are as follows.

(i) Vehicles are equipped with the GPS, which is used for velocity/distance/direction calculation and time synchronization function. They are aware if the moving direction is on an upline or a downline.

(ii) Vehicle communication is assumed on the two-way highway as the system is SRVC system with RSUs installed at a certain interval of distance.

(iii) All RSUs are aware of the pre/post-RSU reaching distance, and intercommunication between them is possible.

In Figure 1, the upper road means a downline while the lower road means an up line. In this case, when the RSU in the left transmits data, almost every vehicle within the transmission range can receive the data. And a vehicle in the upline uses data forwarding mechanism to send data to other vehicle ahead. If vehicles in the upline/downline continue data forwarding onward, possibility of collision may increase due to data transmission by vehicles in the opposite direction. However, if the restricted region on the downline is designated as A and the one on the upline is designated as B, before vehicles in each direction within the restricted region stop data forwarding, almost all of the vehicles can be within the communication coverage without any collision as shown in Figure 1. In addition, even though the SNR is not used as the case with cooperative communication, it is possible to have the advantage of spatial reusability in case of having a relatively high BER.

4. MRB-MAC Protocol

This section presents the design details on the proposed MAC protocol. MRB-MAC is based on the time division multiple access (TDMA) and the IEEE 802.11 PHY. The reason why we select the TDMA instead of the IEEE 802.11p MAC is in order to prevent frame collisions among OBUs during

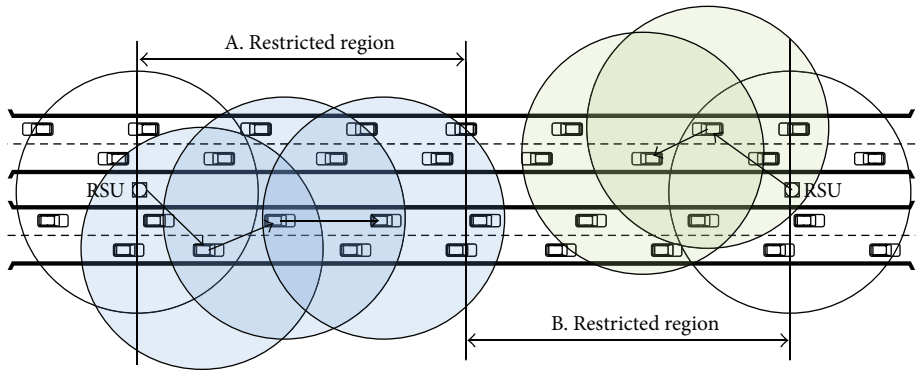


Figure 1: Application scenario.

multihop relay broadcast period. The protocol is composed of two components: RSU's broadcast period and OBU's relay broadcast period. In the RSU's broadcast period, the RSU broadcasts the frames to OBUs. In the OBU's relay broadcast period, the OBUs wait for the delay time calculated by the proposed algorithm for the rebroadcast delay time. Then, the OBU with the shortest delay time rebroadcasts the received data frame.

4.1. Restricted Region. In PVC, the restricted region is used in order to prevent frame collisions between the next RSU and the OBU approaching to the RSU. In addition, the restricted region improves in order to prevent frame collisions between the OBUs on the opposite lane.

Let $\mathrm{TR}_{\mathrm{RSU}}$ and $\mathrm{TR}_{\mathrm{OBU}}$ denote the transmission range of the RSU and OBU. The restricted region presented by PVR is the distance from the RSU location to $3 \times \mathrm{TR}_{\mathrm{RSU}}$ [7]. If the OBUs are in the restricted region, they do not transmit anything. As shown in Figure 2(a), if there is no vehicle in the opposite direction, consideration is taken only into collision-free RSU data frame (RDF) of the RSU. Therefore, the restricted region becomes the distance from the location of the RSU to $2 \times \mathrm{TR}_{\mathrm{RSU}}$. Figure 2(b) shows that a vehicle in the opposite direction leaves from the RSU later than vehicles that move to the right direction. In this case, in order to prevent collision of the OBU's data frame in the opposite direction, the restricted region is set as the distance from the location of the RSU to OBU's $2 \times \mathrm{TR}_{\mathrm{OBU}}$. Lastly, Figure 2(c) shows that vehicles move appropriately on both of the two directions of the road. In this study, the restricted region is designated as the distance from the next location of the RSU to $\mathrm{TR}_{\mathrm{RSU}}$ + half of the distance between the RSUs.

The distance from the point which a vehicle passes the RSU to the predicted point which the vehicle meets another vehicle on opposite side is computed as

$$D = \frac{v_1 \times D_{\mathrm{RSUs}} + \alpha \times v_1 \times v_2}{v_1 + v_2}, \quad \alpha = t_2 - t_1, \tag{1}$$

where v_1 and v_2 are the velocities of vehicles, t_1 and t_2 are the departure time from RSUs, and D_{RSUs} is the distance between RSUs. Then the restricted region arrival distance (RAD) is computed as

$$\mathrm{RAD} = \begin{cases} D_{\mathrm{RSUs}} - 2 \times \mathrm{TR}_{\mathrm{RSU}}, & n = 0, \\ D - 2 \times \mathrm{TR}_{\mathrm{OBU}}, & D > \dfrac{D_{\mathrm{RSU}}}{2}, \\ \dfrac{D_{\mathrm{RSU}}}{2} - \mathrm{TR}_{\mathrm{OBU}}, & \text{otherwise}, \end{cases} \tag{2}$$

where n is the number of the vehicles which depart in the opposite direction from the next RSU. Each RSU computes the RAD using (2) with the information of OBUs received from the next/previous RSUs. Then, this value is set to up/down RR *distance* field when broadcasting the RDF.

4.2. RSU's Broadcast Period. In this period, the RSU broadcasts the RDF periodically, and time slot begins from that point. Figure 3 shows the frame exchange mechanism that takes place after the RSU transmits the RDF. The time slot for the broadcasting of the RDF and the ODF is expressed as T_{RDF} and T_{ODF}, respectively. T_{RDF}, which is the time slot period of the RDF, is calculated as follows:

$$T_{\mathrm{RDF}} = T\,(\mathrm{RDF}) + \mathrm{Max\ \ prop.\ delay} + \mathrm{SIFS}, \tag{3}$$

where T(RDF) means the time that is taken for transmitting the RDF. And T_{ODF} is calculated as follows:

$$T_{\mathrm{ODF}} = T_{\mathrm{RDF}} + \mathrm{Maximum\ delay\ time}. \tag{4}$$

Then, the RSU's broadcasting interval time is $T_{\mathrm{RDF}} + 3 \times T_{\mathrm{ODF}}$.

Figure 4 shows frame format of the RDF. In the figure, *lane count* field means the one-way maximum lane count of a highway where the RSU is installed. *Time stamp* field means the time when the RSU has broadcast, while the OBU uses the field to calculate the time slot period of the TDMA. RSU *location* field means the location of the RSU that transmits the RDF, while up/down *RR distances* field means the reaching distance up to the restricted region (RR) of up/downline, respectively. The OBU that receives the RDF uses a restricted region distance that is required according to the moving direction.

4.3. OBU's Relay Broadcast Period. The OBU that received the RDF uses GPS device to determine if the OBU is on an upline

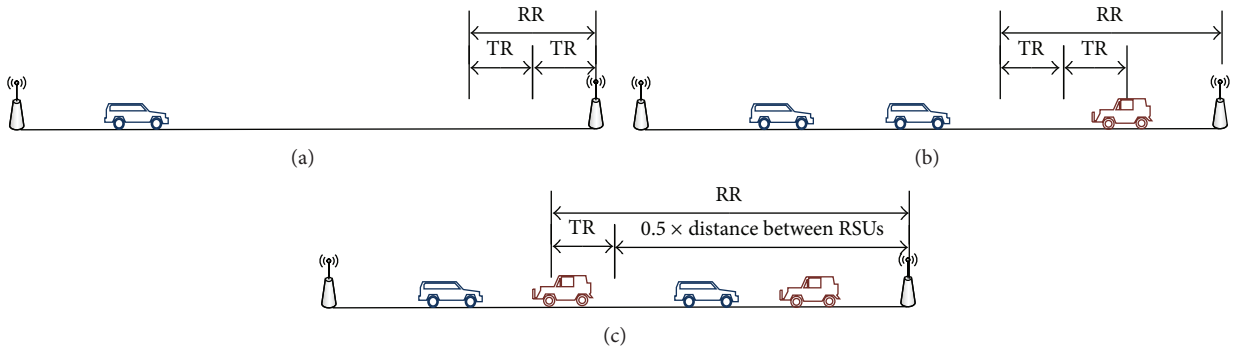


Figure 2: Restricted region examples.

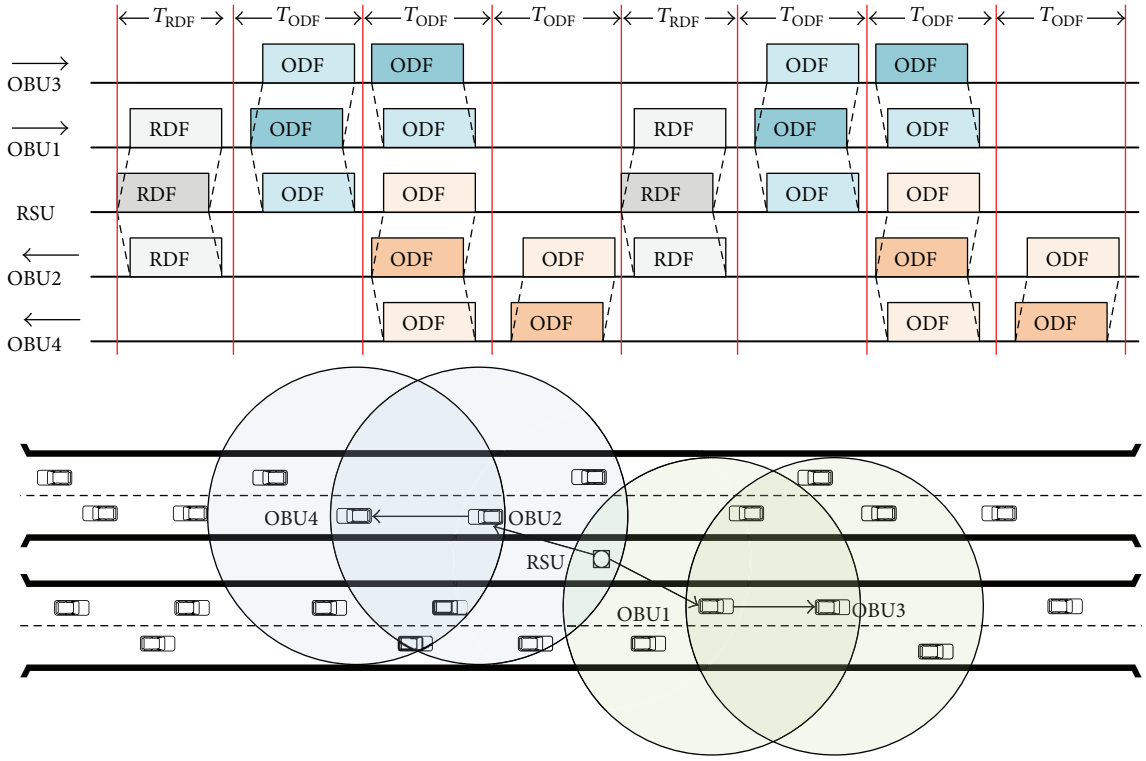


Figure 3: Frame exchange mechanism in the RSU/OBU's broadcast periods.

RSU data frame (RDF) (type: 1)

	Type	Lane count	Time stamp	RSU location	Up/down RR distances	Length	Data	CRC
Octets:	1(4 bits/4 bits)		8	8	4(2/2)	2	Variable	4

OBU data frame (ODF) (type: 2)

	Type	Lane count	Time stamp	OBU location	Direction	Velocity	RR distance	Length	Data	CRC
Octets:	1(4 bits/4 bits)		8	8	1	1	2	2	Variable	4

Figure 4: Frame format in the RSU/OBU's broadcast periods.

or a downline before performing the proposed rebroadcast mechanism. If the OBU is on an upline, it performs the rebroadcast mechanism immediately after the time of T_{RDF} has passed since the time point of RDF's *time stamp* field value just as the OBU1 in Figure 3. If the OBU is on a downline, it waits for the time of T_{ODF} before performing the mechanism. This prevents the ODFs that were broadcasted by the OBUs that receive RDF on the up/downline from colliding in the RSU. The RSU calculates the restricted region based on location and velocity of the ODF that it received.

The values of *time stamp* field and RR *distances* field of an ODF shown in Figure 4 are identical to the field values in the RDF that an OBU received. Based on this fact, if the OBU receives the ODF, procedures as below are taken.

(i) If the OBU receives the ODF that does not have the recent *time stamp* field value or the ODF of an OBU in the opposite direction, the OBU discards it.

(ii) If the OBU is within the restricted region, it does not perform the rebroadcast mechanism.

(iii) Otherwise, the OBU performs the rebroadcast mechanism.

(iv) If a carrier sensing occurs during the rebroadcast mechanism of the OBU, the mechanism stops.

4.4. Rebroadcast Mechanism. The key to the proposed rebroadcast mechanism is the RDF/ODF rebroadcast delay time algorithm (RDTA). The RDF RDTA is used by the OBU that received the RDF to calculate delay time. The direction and location of the OBU1 are used to induce a linear equation in the two variables while the location and transmission range of the RSU can be used to induce a circle's equation as shown in Figure 5(a). The RSU should be installed to ensure that the road traverses the transmission range of the RSU so that the two equations have the two solutions. And one of the two solutions becomes close according to the direction that the OBU moves, which becomes the RSU departure location that is the boundary that determines if it goes out of the transmission range of the RSU. For example, the departure distance between the OBU's current location and the RSU departure location is OBU1's D1 and OBU2's D2 in Figure 5(a). And the OBU calculates the RSU departure time (RDT) based on

$$\text{RDT} = \frac{\text{OBU's departure distance}}{\text{OBU's velocity}}. \tag{5}$$

According to [15, 16], a safe headway is at least 2 seconds or more because it is impossible to follow a vehicle safely with a headway less than 2 seconds. For example, the safe headways of 100 km/h and 80 km/h are 54 m and 44 m, respectively. Let T_{safe} denotes the safe headway (second). Then, when the OBU receives the RDF, its delay time value (DTV) is computed as

$$\text{DTV}_{\text{RDF}} = \begin{cases} \text{MDT} \times \dfrac{\text{RDT}}{T_{\text{safe}}}, & 0 \le \text{RDT} \le T_{\text{safe}}, \\ \infty, & \text{otherwise}, \end{cases} \tag{6}$$

where MDT is the maximum delay time. If the RDT of the OBU is smaller than T_{safe}, this means that the probability that there is no vehicle in front of it in the TR_{RSU} area is very high. This reduces the number of competing OBUs, which results in reduction of collision probability.

The ODF RDTA is used to calculate the delay time of the OBU that received the ODF. While most of the related research works propose the algorithms based on the distance, the proposed RDTA uses the velocity and angle as well as the distance. Using the safe headways, the vehicles that received the ODF compute the number of vehicles in front of them in the same lane in the transmission range by

$$\text{VC} = \left\lfloor \frac{\text{TR}_{\text{OBU}} - D_{\text{OBUs}}}{\text{Velocity}_{\text{OBU}} \times s_{\text{headway}}} \right\rfloor, \tag{7}$$

where the unit of $\text{Velocity}_{\text{OBU}}$ is "m/s", and D_{OBUs} is the distance between OBUs, as shown in Figure 5(b). And the OBU that received the ODF calculates the lane number (LN) of the received OBU in

$$\text{LN} = \left\lfloor \frac{\left| D_{\text{OBUs}} \times \sin\left(\theta_{\text{OBUs}}\right) \right| - L_{\text{width}}/2}{L_{\text{width}}} \right\rfloor, \tag{8}$$

where D_{OBUs} is the distance between OBUs, L_{width} is the width of a lane, θ_{OBUs} is the angle between the direction of the transmitting OBU and the location of the OBU that received the ODF as shown in Figure 5(b). For example, if the value of LN is 0, this means that the OBU is in the same lane as that of the transmitting OBU. If the value of LN is 2, this means that the OBU is in the second lane above or below. In Figure 5(b), the OBU can tell if it is in the lane above or below the transmitting OBU because $\sin(\alpha)$ is larger than 0, and $\sin(\beta)$ is smaller than 0. However, even though a vehicle is equipped with a GPS device, it is difficult to find out which lane it is on. Therefore, in the proposed algorithm, the OBU that received the ODF based on the transmitting OBU uses the following to calculate its virtual lane number (VLN):

$$\text{VLN} = \begin{cases} 2 \times \text{LN}, & \sin(\theta) \le 0, \\ 2 \times \text{LN} + 1, & \text{otherwise}. \end{cases} \tag{9}$$

For example, when D, E, and F in Figure 6(a) receive the ODF that A transmitted, H in Figure 6(b) becomes the transmitting OBU while the receiving OBUs are K, L, and M, respectively. As a result, the VLNs of K, L, and M are 0, 2, and 4, respectively. In the same way, when C in Figure 6(a) transmits the ODF, I, J, and K in Figure 6(b) receive the ODF while the *VLN*s are 3, 1, and 0, respectively.

The ODF delay time value (DTV) is computed using the results of (7) and (9) as

$$\text{DTV}_{\text{ODF}} = T_{\text{tick}} \times \text{VC} \times \text{MLC} + T_{\text{tick}} \times \text{VLN}, \tag{10}$$

where MLC is the maximum virtual lane count, and T_{tick} means the minimum time unit. This provides that if a driver keeps the safe headway, then DTV_{ODF} has a unique value and the collision-free communication becomes possible. If DTV_{ODF} is larger than the maximum delay time, the OBU that received the ODF stops the rebroadcast mechanism.

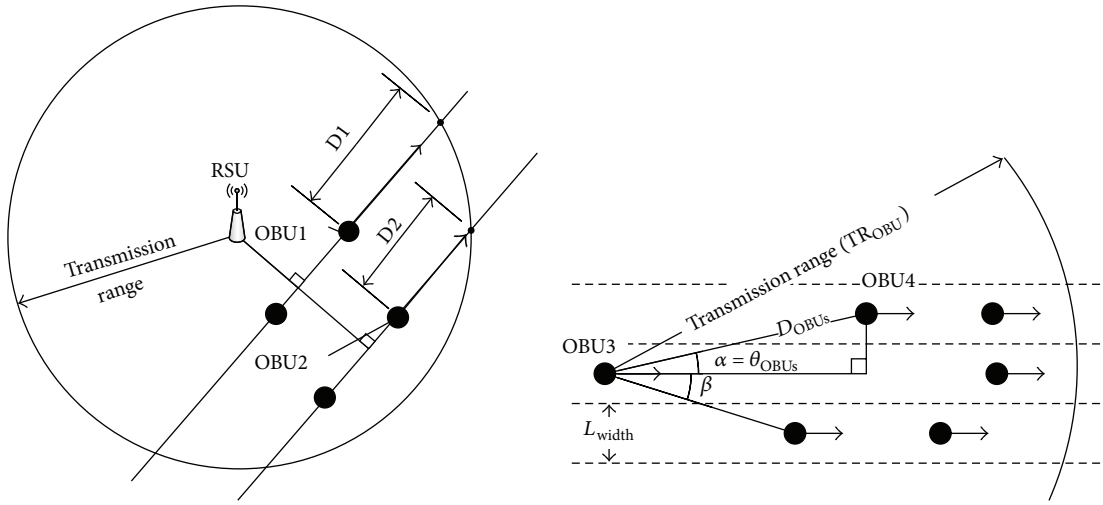


Figure 5: Examples for the rebroadcast delay time algorithm.

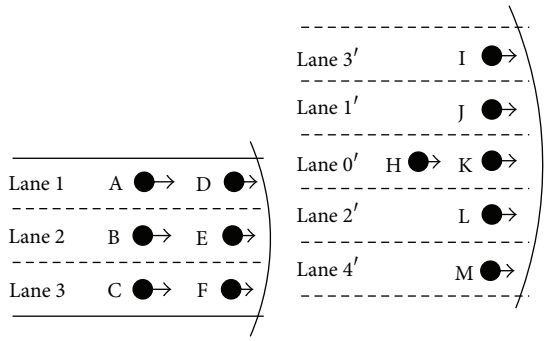


Figure 6: Example for the lane number calculation.

Table 1: Simulation parameters.

Parameter	Value
Maximum delay slot time count	15
Lane width (m)	4
Safe headway (second)	2
RSU count	2
Simulation time (second)	150

5. Simulation Results

In this section, we show the performance evaluation of the proposed MRB-MAC protocol. The simulation experiments are conducted by ns-2 with version 2.35-RC7 [17].

The main parameters of our simulations are shown in Table 1. The physical layer model is a two-ray ground inflection model, and the application is set to be a constant bit rate. In order to set up the other parameters, we refer to "tcl/ex/802.11/broadcast_validation.tcl" file in ns-2.

The deployment of the vehicles is assumed to be uniform because they move on highways. The number of the vehicles decreases as the distance between them increases when the speed of the vehicles becomes high with the safe headway being the same. The time available for communication with RSU increases as their speed becomes low.

Figure 8 shows the aggregated throughputs of MRB-MAC and the no-relay, which received only RSU frames, by the velocity with the distance between RSUs being 1 km (scenario: Figures 7(b) and 7(c)). From the result, as the velocity of a vehicle increases, the service time for downloading decreases, which leads to the decrement of the throughput.

Figure 9 shows the changes of the aggregated throughput of MRB-MAC as the distance between the RSUs changes at 4-lane roadway with 80 km/h velocity (scenario: Figure 7(c)). From the figure, it can be known that the throughput reduces sharply at the distance between the RSUs of 500 meters and 600 meters. This is because the interference occurs when moving vehicles in two-way directions relay the packets with the maximum sensing range being approximately 220 meters in the simulation configuration.

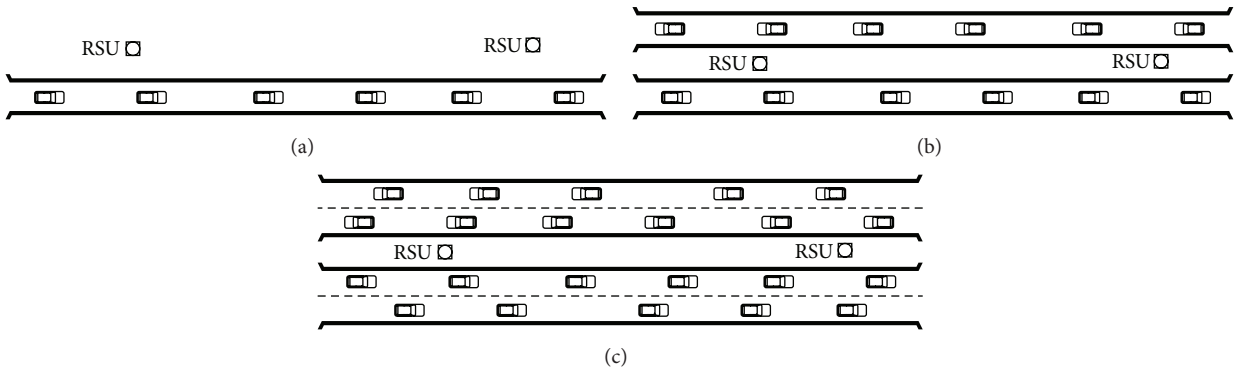


Figure 7: Simulation scenarios.

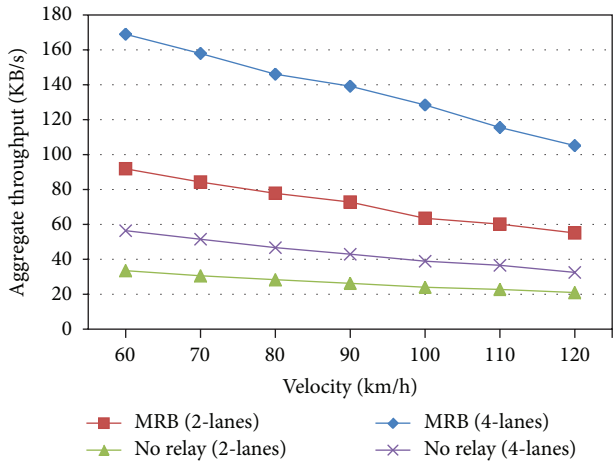


Figure 8: The aggregate throughput by the vehicle's velocity.

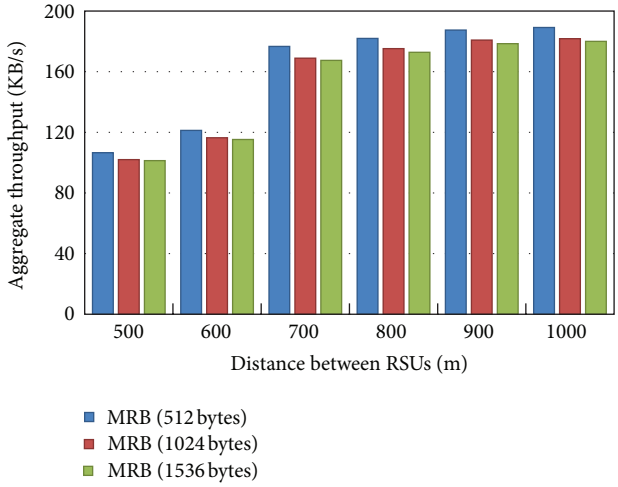


Figure 9: The aggregate throughput by the distance between RSUs (4 lanes, 80 km/h).

Figure 10 shows the results with the only change in 1-lane roadway from 4-lanes (scenario: Figure 7(a)). In the results, however, it is checked that the sharp reduction of the

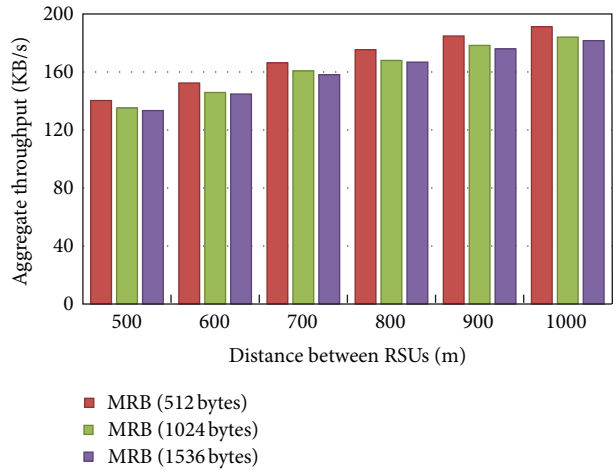


Figure 10: The Aggregate throughput by the distance between RSUs (1-lane, 80 km/h).

throughputs does not occur at the distance between the RSUs of 500 meters, because the vehicles move in one direction.

Figure 11 shows that the average delay increases as the packet length does (scenario: Figure 7(a)). It is known that as the increment of the data packet length makes both T_{RDF} and T_{ODF} longer, too big packet makes the interval of RSU receiving RDF take longer as well, which in turn leads to the performance degradation.

6. Conclusion

In this paper, we have proposed an energy-efficient multihop relay broadcast MAC protocol, which is designed for RSU downloading service in RVC systems. The RVC system has a disadvantage that its service coverage is small. In order to overcome it, we first present the restricted region for collision-free communication among RSUs and OBUs. Then, we design a protocol with a rebroadcast mechanism using a vehicle's velocity, distance, and angle from nearby vehicles. Simulation results show that the proposed protocol provides the expanded service coverage with energy-efficient collision-free communication.

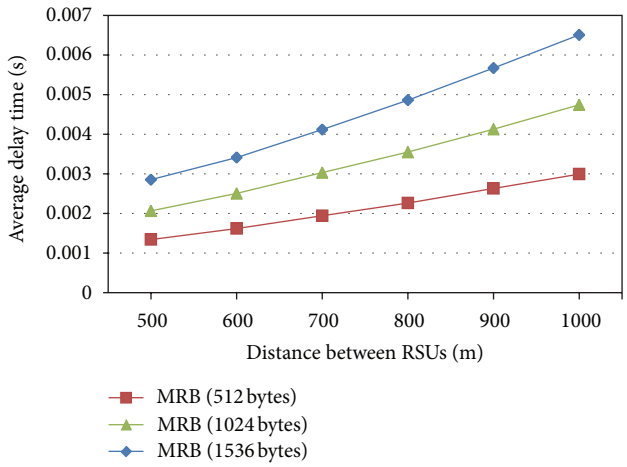


Figure 11: The average delay time by the distance between RSUs (1-lane, 80 km/h).

A disadvantage of our protocol is that network connectivity may not be guaranteed in a scenario with low vehicle density. However, by adjusting the distance between RSUs adequately depending on traffic volumes when an RSU is installed under a SRVC system, our protocol is expected to provide the good performance. In the future, we will focus on the cooperative communication in the urban scenario as well as the highway scenario.

Acknowledgment

The present research has been conducted by the research grant of Kwangwoon University in 2011.

References

[1] M. Sichitiu and M. Kihl, "Inter-vehicle communication systems: a survey," *IEEE Communications Surveys and Tutorials*, vol. 10, no. 2, pp. 88–105, 2008.

[2] T. Willke, P. Tientrakool, and N. Maxemchuk, "A survey of inter-vehicle communication protocols and their applications," *IEEE Communications Surveys and Tutorials*, vol. 11, no. 2, pp. 3–20, 2009.

[3] S. Prahmkaew, "Performance evaluation of convergence Ad Hoc networks," *Journal of Convergence*, vol. 1, no. 1, pp. 101–106, 2010.

[4] N. Kryvinska, D. van Thanh, and C. Strauss, "Integrated management platform for seamless services provisioning in converged network," *International Journal of Information Technology, Communications and Convergence*, vol. 1, no. 1, pp. 77–91, 2010.

[5] J. Zhang, Q. Zhang, and W. Jia, "VC-MAC: a cooperative MAC protocol in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 3, pp. 1561–1571, 2009.

[6] T. Zhou, H. Sharif, M. Hempel, P. Mahasukhon, W. Wang, and T. Ma, "A novel adaptive distributed cooperative relaying MAC protocol for vehicular networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 1, pp. 72–82, 2011.

[7] M. F. Jhang and W. Liao, "On cooperative and opportunistic channel access for vehicle to roadside (V2R) communications," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '08)*, pp. 5053–5057, December 2008.

[8] G. Zhao and A. Kumar, "Lifetime-aware geographic routing under a realistic link layer model in wireless," *International Journal of Information Technology, Communications and Convergence*, vol. 1, no. 3, pp. 297–317, 2011.

[9] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM '00)*, pp. 243–254, August 2000.

[10] O. K. Tonguz, N. Wisitpongphan, J. S. Parikh, F. Bai, P. Mudalige, and V. K. Sadekar, "On the broadcast storm problem in ad hoc wireless networks," in *Proceedings of the 3rd International Conference on Broadband Communications, Networks and Systems (BROADNETS '06)*, pp. 1–11, October 2006.

[11] M. Heissenbüttel, T. Braun, M. Wälchli, and T. Bernoulli, "Optimized stateless broadcasting in wireless multi-hop networks," in *Proceedings of the IEEE 25th Conference on Computer Communications (INFOCOM '06)*, pp. 23–29, Barcelona, Spain, 2006.

[12] S. Bai, Z. Huang, and J. Jung, "Beacon-based cooperative forwarding scheme for safety-related inter-vehicle communications," in *Proceedings of the International conference on Computational Science and Its Applications (ICCSA '10)*, vol. 6019 of *Lecture Notes in Computer Science*, pp. 520–534, 2010.

[13] P. Liu, Z. Tao, and S. Panwar, "A cooperative MAC protocol for wireless local area networks," in *Proceedings of the IEEE International Conference on Communications (ICC '05)*, pp. 2962–2968, May 2005.

[14] I. Tal and G.-M. Muntean, "User-oriented cluster-based solution for multimedia content delivery over VANETs," in *Proceedings of the IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB '12)*, June 2012.

[15] M. Fiorani, M. Mariani, F. Tango, and A. Saroldi, "SASPENCE—safe speed and safe distance," in *Proceedings of the 5th European Congress on ITS*, Hannover, Germany, 2005.

[16] C. J. Colbourn, I. D. Brown, and A. K. Copeman, "Drivers' judgments of safe distances in vehicle following," *Human Factors*, vol. 20, no. 1, pp. 1–11, 1978.

[17] NS-2, "Network simulator-2," http://www.isi.edu/nsnam/ns.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 514037, 10 pages
http://dx.doi.org/10.1155/2013/514037

Research Article

Repeated Game-Inspired Spectrum Sharing for Clustering Cognitive Ad Hoc Networks

Cuiran Li[1,2] and Jianli Xie[1,2]

[1] *School of Electronics and Information Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China*
[2] *State Key Lab. of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China*

Correspondence should be addressed to Jianli Xie; xiejl@mail.lzjtu.cn

Received 6 October 2012; Revised 15 January 2013; Accepted 28 January 2013

Academic Editor: Jiman Hong

The paper studies the cooperative spectrum sharing among multiple secondary users (SUs) in a clustering cognitive ad hoc network. The problem is formulated as a repeated game with the aim of maximizing the total transmission rate of SUs. Firstly, a clustering formation procedure is proposed to reduce the overhead and delay of game process in cognitive radio network (CRN). Then the repeated game-inspired model for SUs is introduced. With the model, the convergence condition of the proposed spectrum-sharing algorithm is conducted, and the convergence performance is investigated by considering the effects of three key factors: transmission power, discount factor, and convergence coefficient. Furthermore, the fairness of spectrum sharing is analyzed, and numerical results show a significant performance improvement of the proposed strategy when compared to other similar spectrum-sharing algorithms.

1. Introduction

The scarcity of spectrum has become a major bottleneck of the development of next generation wireless communication system. Cognitive radio (CR), which allows unlicensed or secondary users (SUs) to share the spectrum with licensed or primary users (PUs), shows great promise to enhance the spectrum utilization efficiency [1, 2]. The CR technology enables the SUs to opportunistically access the available spectrum bands through four main functionalities: spectrum sensing, spectrum managing, spectrum mobility, and spectrum sharing [3]. Among these, spectrum sharing is one of the most important functions in cognitive radio, which allows SUs to share the available spectrum bands among the coexisting PUs [4]. It is essential for improving spectrum utilization.

There exist some research efforts on the problem of spectrum sharing in CR. Among these studies, a centralized spectrum management scheme was proposed in [4]. It greatly improves the system performance over the (iterative water-filling) IWF scheme by utilizing a centralized spectrum management center (SMC). However, due to the heterogeneous and dynamic nature of cognitive radio, centralized approach is not practical. Instead, in some studies, the distributed approach which does not need any central controller is suggested [5]; in [5], asynchronous distributed pricing scheme is proposed, based on the signal exchange via coordination between users to compensate the ascendant interference level. Distributed approach provides the better adaptation capability to CR in the dynamically changing heterogeneous environment, but the coordination among SUs results in significant amount of coordination delay. In [6], a novel distance-dependent MAC protocol for CRN is proposed, which attempts to maximize the CRN throughput. Regretfully, these protocols do not consider the fairness of the spectrum sharing.

Game theory, which analyzes the conflict and cooperation among decision makers (users), has widely used in designing efficient spectrum sharing. A dynamic game model is presented in [7], in which the SUs can iteratively adapt their strategies in terms of requested spectrum size. The stability condition of the dynamic behavior for the spectrum-sharing scheme is investigated. In [8, 9], the authors investigate whether spectrum efficiency and fairness can be obtained by modeling the spectrum sharing as a repeated game. In [10], the authors model the channel assignment and power

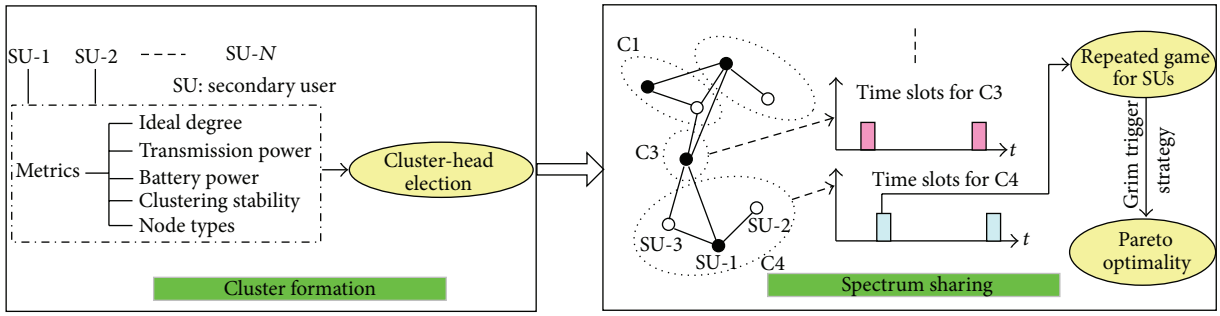


Figure 1: Repeated game-inspired spectrum-sharing model.

control problems as a noncooperative game, in which all wireless users jointly pick an optimal channel and power level to minimize a joint cost function. A no-regret learning algorithm using the correlated equilibrium concept to coordinate the secondary spectrum access is considered in [11]. In [12], a self-enforcing truth-telling game mechanism is used to suppress cheating and collusion behavior of selfish users for spectrum sharing. It is shown that the SUs can get the highest rate utility only by announcing their true private information under the assumption that all the SUs have the same maximal transmission power. Based on the Nash bargaining in cooperative game, an improved utility function is proposed in [13] to maximize the profit product of all the SUs.

Consider that the CRN is characterized by a lack of centralized control and the restriction that global information is not available, which requires that the game algorithms for spectrum sharing should be completely distributed relying on local information. It motivate us to employ local interaction games [14], which have been recently introduced in CRN research known as graphical games in [15]. In [16], two cases of local interaction game are proposed to cope with the lack of centralized control and local influences. The first is local altruistic game, in which each user considers the payoffs of itself as well as its neighbors rather than considering itself only. The second is local congestion game, in which each user minimizes the number of competing neighbors. It is shown that, with the local games, global optimization is achieved with local information. Although some progresses have been achieved in the above approaches, the problem of spectrum sharing more efficient and fair is not yet solved.

In this paper, a repeated game theoretic model is proposed, with the aim to maximize the total rate revenue of SUs in cognitive ad hoc network. Clustering is executed firstly in the model to avoid frequent collisions and to reduce the coordination delay between SUs. With the model, the convergence condition for the total revenue maximum is studied. The transmission power, discount factor, and convergence coefficient, which impact the convergence behavior, are explored. Besides, the fairness of spectrum sharing is investigated.

The rest of the paper is organized as follows. In Section 2, the system model is introduced. In Section 3, the total rate revenue of the spectrum-sharing algorithm, the convergence behavior and fairness of the algorithm are analyzed. In Section 4, the performances of the rate revenue, convergence, and fairness are simulated and evaluated. Finally, conclusions are drawn in Section 5.

2. System Model and Descriptions

Spectrum-sharing model based on repeated game theory is presented in Figure 1. In the scenario, N secondary users compete to access the available spectrum to transmit data. Note that the wireless channels are assumed to be quasistatic for each time slot, that is, channels remain unchanged within the time slot duration, but they vary from one slot to another one.

The model consists of two parts: cluster formation and spectrum sharing. With no clustering in a cognitive network, the collision happened as long as L ($L \geq 2$) SUs in the transmission range of each other have data to transmit simultaneously [17]. By introducing the idea of clustering in the network, ordinary (cluster-member) SU communicates with only cluster-head (CH) SU, and collisions between SUs can be decreased greatly. A combined weight metrics to elect the CHs is considered in the cluster formation process.

For the spectrum sharing, a static repeated game among SUs can be used to obtain the Pareto optimality in a clustering fashion by the "grim trigger" strategy.

Definition 1. The Nash equilibrium is a set of strategies, one for each player, such that no player has incentive to unilaterally change her action. Players are in equilibrium if a change in strategies by any one of them would lead that player to earn less than if she remained with her current strategy. For games in which players randomize (mixed strategies), the expected, or average payoff (also termed as revenue, utility or outcome) must be at least as large as that obtained by any other strategy.

Definition 2. The Pareto optimality is a measure of efficiency. An outcome of a game is Pareto optimal, if there is no other outcome that makes every player at least as well off and at least one player strictly better off. That is, a Pareto optimal outcome cannot be improved upon without hurting at least one player. Often, a Nash Equilibrium is not Pareto optimal implying that the players' payoffs can all be increased.

In order to stimulate cooperation among selfish players (SUs) and achieve the Pareto optimality, the "grim trigger" strategy is adopted in our spectrum-sharing model. The "grim trigger" is a trigger strategy employed in a repeated game [18]. Initially, a player using grim trigger will cooperate, as soon as the opponent defects (thus satisfying the trigger condition), and the player using grim trigger will defect for the remainder of the iterated game. Since a single defect by the opponent triggers defection forever, grim trigger is the most strictly unforgiving of strategies in an iterated game.

2.1. Clustering Formation. In this section, a clustering procedure that exploits combined weight metrics is proposed. The idea of using combined weight metrics, including the ideal degree, transmission power, and battery power, has been considered in the literature [19]. In this paper, in addition to the above weight metrics, the clustering stability and node type have also been considered as weight metrics to elect the cluster-heads (CHs). The clustering stability is used to retain the stability of the network topology, and node type helps to reflect the realistic characteristics of multiple types of nodes in cognitive ad hoc networks.

The network formed by the SUs and transmission links can be represented by an undirected graph $G = (V, E)$, where V represents the set of SUs and E represents the set of links e_i. Clustering can be thought as a graph-partitioning problem with some constraints. Look for the set of vertices $S \subseteq V(G)$, such that

$$\bigcup_{v \in S} N[v] = V(G), \tag{1}$$

where $N[v]$ is the neighborhood of SU-v. The set S is called a dominating set such that every vertex of G belongs to S or has a neighbor in S. The dominating set of the graph is the set of CHs. It might be possible that a node is physically nearer to a CH but is the member of another CH.

The following metrics are considered in our clustering procedure for a cognitive ad hoc network.

(i) *Ideal Degree*. Each CH can ideally support d_{ideal} (a predefined threshold) nodes to ensure that CHs are not overloaded, and the efficiency of the system is maintained at the expected level. If the CH tries to serve more nodes than it is capable of, the system efficiency decreases in the sense, because the nodes have to wait longer for their turn to get the share of the resource. A high system throughput can be achieved by limiting or optimizing the degree of each CH.

The degree difference from the ideal degree helps in efficient MAC functions and load balancing because it is always desirable for a CH to handle up to a certain number of nodes in its cluster.

The neighbors of each SU-v (i.e., SUs within its transmission range) is defined as the degree of node v, d_v

$$d_v = |N(v)| = \sum_{v' \in V, v' \neq v} \left\{ \text{dist}\left(v, v'\right) < tx_{\text{range}} \right\}. \tag{2}$$

The degree difference Δ_v for every node v is computed as

$$\Delta_v = \left| d_v - d_{\text{ideal}} \right|, \tag{3}$$

where d_{ideal} is the number of nodes that a CH can handle ideally.

(ii) *Transmission Power*. It is known that more power is required to communicate to a larger distance. As the nodes move away from the CH, the communication may become difficult due mainly to signal attenuation with increasing distance.

The usual attenuation in the signal strength is inversely proportional to some exponent of the distance, which is usually approximated to 4 in cellular networks, where the distance between mobiles and base stations is of the order of 2-3 miles. In ad hoc networks, the distances involved are rather small (approximately hundreds of meters). In this range, the attenuation can be assumed to be linear [20].

For every node, the sum of the distances, D_v, with all its neighbors can be computed:

$$D_v = \sum_{v' \in N(v)} \left\{ \text{dist}\left(v, v'\right) \right\}. \tag{4}$$

(iii) *Battery Power*. The battery power can be efficiently used within certain transmission range; that is, it takes less power for a node to communicate with other nodes if they are within close distance to each other. A CH consumes more battery power than an ordinary node, since it has extra responsibilities to carry out for its members.

We consider a heterogeneous network with multiple initial energy levels. The cumulative time, P_v, during which an SU-v acts as a CH, implies how much battery power has been consumed.

(iv) *Clustering Stability*. In order to avoid frequent CH changes, it is desirable to elect a CH that does not move very quickly. The focus of the most existing literatures has mostly been on absolute mobility of the nodes without taking into consideration the relative mobility. Thus, the stability of the network is perturbed. Our clustering procedure achieves the network stability by considering the relative mobility of nodes.

The average distance for every node v from its neighbor u till current time T is calculate as

$$\overline{d_{vu}} = \frac{1}{T} \sum_{t=1}^{T} d_{vu}^t, \tag{5}$$

where d_{vu}^t is the distance between node v and u at time t.

Let LS_{vu} represent the link stability between v and u, expressed by

$$\text{LS}_{vu} = \frac{1}{T} \sum_{t=1}^{T} \left(d_{vu}^t - \overline{d_{vu}} \right)^2. \tag{6}$$

The average of the link stability for every node v, with all its neighbors [21], is given by

$$\text{LStab}_v = E\left(\text{LS}_{vu} \mid u \in N(v)\right) = \frac{1}{d_v} \sum_{u \in N(v)} \text{LS}_{vu}. \tag{7}$$

The cluster stability can be obtained as

$$\text{CStab}_v = \frac{1}{d_v} \sum_{u \in N(v)} E\left(\text{LS}_{vu} - \text{LStab}_v\right)^2. \tag{8}$$

From (8), we can see that the CStab_v is somewhat like that of variance, which reflects the relative mobility of the nodes.

(v) *Node Types*. In many realistic ad hoc networks, multiple types of nodes do coexist [22]. For example, in a battlefield network, portable wireless devices are carried by soldiers, and more powerful and reliable communication devices are carried by vehicles, tanks, aircrafts, and satellites; these devices/nodes have different communication characteristics in terms of transmission power, data rate, processing capability, reliability, security level, and so forth.

In heterogeneous ad hoc networks, it would be more realistic to elect CHs for considering the different types of nodes. For simplicity, two types of nodes are considered in the network. One type of node has larger transmission range (power) and data rate and better processing capability and is more reliable and robust than the other types. Accordingly, the mapping values of the two types of nodes T_v might be denoted by 1 and 2.

The combined weight W_v for each SU-v can be calculate by

$$W_v = w_1\Delta_v + w_2 D_v + w_3 P_v + w_4 \mathrm{CStab}_v + w_5 T_v. \quad (9)$$

Subject to:

$$w_1 + w_2 + w_3 + w_4 + w_5 = 1, \quad (10)$$

where w_1, w_2, w_3, w_4, and w_5 are the weighing factors for the corresponding metrics.

The contribution of the individual metrics can be tuned by choosing the appropriate combination of the weighing factors [21]. The node with the smallest W_v would be selected as the CH. All the neighbors of the chosen CH are no longer allowed to participate in the election procedure. The clustering procedure continues until the remaining nodes are selected as CHs or assigned to a cluster.

The first component in (9), Δ_v, contributing towards the combined metric W_v helps in efficient MAC functioning because it is always desirable for a CH to handle up to a certain number of nodes in its cluster. The motivation of D_v is mainly related to energy consumption. A CH is able to communicate better with its neighbors having closer distances from it within the transmission range. The third component, P_v, is measured as the total (cumulative) time a node acts as a CH. As a heterogeneous network with multiple initial energy levels is considered, the power currently available at the node depends on the node's initial power, the actual network traffic, and the length of the links. The component of CStab_v is measured as the clustering stability, which is mainly related to the velocity and direction of the mobile nodes, especially the mobility relative to CHs. The nodes' association and dissociation to and from clusters perturb the stability of the network, and thus reconfiguration of CHs is unavoidable. It is desirable to elect a CH that does not move very quickly relative to its neighbors. The last component T_v is related to the types of nodes. The more powerful, the more responsibility for communication nodes must be taken. As a result, it would be more appropriate for a node with more powerful capacity to be a CH.

Table 1: Execution of the clustering strategy.

Node ID	Δ_v	D_v	P_v	CStab_v	T_v	W_v
1	1	3	1	3	2	1.85
2	0	6	2	3.8	1	1.84
3	1	3	2	3.6	2	1.93
4	1	9	4	4.13	2	3.26
5	1	3	2	3.81	2	1.94
6	0	6	0	2.92	1	1.7
7	0	7	3	3.54	2	2.42
8	0	7	2	3.73	1	2.04

The proposed clustering strategy is demonstrated with the help of Figures 2(a)–2(d). All numeric values obtained from clustering process are given in Table 1. Figure 2(a) shows the initial configuration of the nodes (SUs) in a cognitive radio network, where an edge (link) between two nodes in the figure signifies that the nodes are neighbors of each other, and the length of a link represents the distance of two nodes. In the figure, the degree difference, Δ_v, of each node with ideal node degree $d_{\mathrm{ideal}} = 2$ is computed. The arrows in Figure 2(b) represent the speed and direction of movement associated with every node. A longer arrow represents faster movement, and a shorter arrow indicates slower movement. Some arbitrary values for P_v are chosen which represent the amount of time a node has acted as a CH. The values for T_v are chosen randomly. If $T_v = 1$, it implies that a node is more reliable and robust than the nodes whose $T_v = 2$. The weighting factors are chosen with satisfying (10). The weighting factors considered are $w_1 = 0.35$, $w_2 = 0.2$, $w_3 = 0.05$, $w_4 = 0.05$, and $w_5 = 0.35$. Figure 2(c) shows how a node with minimum W_v is selected as the CH, where the pink solid nodes represent the CHs elected for the network. Figure 2(d) shows the initial clusters formed by execution of our clustering strategy and the achieved connectivity in the network, where a dashed ellipse is used to express a cluster. We can see that no two CHs are immediate neighbors, since all the neighbors of the chosen CH belong to the same cluster. The network connectivity is achieved through the higher power transmission range of CHs. Also, it can be noted that a single component graph is obtained in this case which means that there is a path from a node to any other node. For simplicity, the ideal node degree is set to 2 in this paper. Without loss of generality, the ideal node degree can be set as an arbitrary positive integer.

2.2. Repeated Game-Based Spectrum Sharing. The SUs sharing the spectrum of licensed users (PUs) may lead to the following conflicting problems: (i) limitation on the transmission power in each channel for minimum interference to coexisting PUs and (ii) certain signal-to-noise ratio (SNR) required for data transmission of SUs without substantial performance degradation. Cooperation among SUs has been proved to be beneficial in solving such conflicting interests [23]. Such cooperation can be achieved with the application of the concepts of game theory. But, cooperative game faces scalability problem to be implemented in CR networks. In a

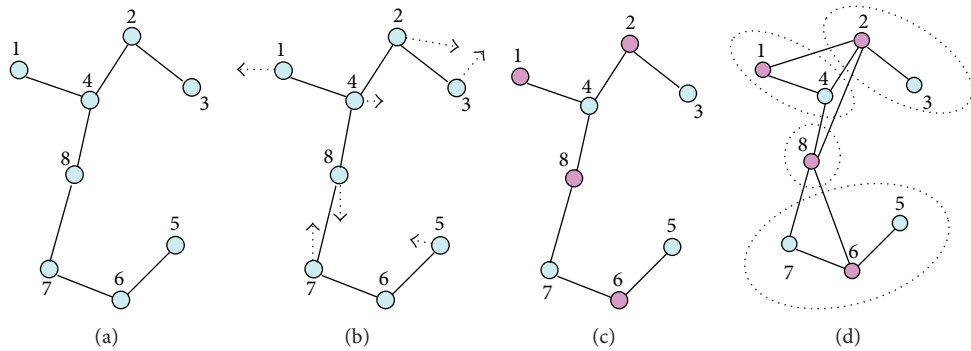


Figure 2: (a) Initial configuration of nodes and neighbors identified. (b) Velocity of the nodes. (c) CHs identified. (d) Clusters identified and connectivity achieved.

large CR network, the overhead and delay of the game process will be unbearable, if all the SUs play a single game; on the other hand, it is not reasonable to let SUs that are set apart far way (thus has little direct mutual impacts) play the same game. So it is necessary and reasonable to group the cognitive radios network into multiple clusters firstly.

In order to model and analyze long-term interactions among players, the repeated game model is used where the game is played for multiple stages. A repeated game is a special form of an extensive-form game in which each stage is a repetition of the same strategic-form game. Particularly, the spectrum-sharing problem can be modeled as the outcome of a repeated game, in which the players are the SUs their strategies (actions) are the choice of spectrum resources.

In a repeated game, a normal-form game is mathematically defined as

$$A = \left\{V, \{B_v\}_{v\in V}, \{U_v\}_{v\in V}\right\}, \tag{11}$$

where V is the finite set of SUs and B_v is the set of strategies associated with SU-v. Define $\mathbb{B} = \times B_v v \in V$ as the strategy space and U_i: $\mathbb{B} \rightarrow \mathbb{R}$ as the set of utility functions that the SUs associate with their strategies. For every SU-v in game A, the utility function, U_v, is a function of b_v, the strategy selected by SU-v, and of the current strategy profile of its opponents: b_{-v}.

In analyzing the outcome, as the decision of one SU is influenced by the other SUs' decisions, we are interested to determine, if there exists a convergence point, that is, Nash equilibrium (NE), for the spectrum-sharing algorithm, from which no SU would deviate anymore. A strategy profile for the SUs, $B = [b_1, b_2, \ldots, b_V]$, is an NE if and only if

$$U_v\left(b_v, b_{-v}\right) \geq U_v\left(b_v', b_{-v}\right), \quad \forall v \in V,\ b_v' \in B_v. \tag{12}$$

If the equilibrium strategy profile in (12) is deterministic, a pure strategy NE exists.

When there is more than one NE in the repeated game process, it is natural to ask whether there exists an optimal one, that is, Pareto optimality. In order to stimulate cooperation among selfish SUs and achieve the optimality, the "grim trigger" strategy is adopted. For the case of no deviation from cooperation in a repeated game, the utility function at every stage for SU-v is unchangeable. The overall utility for SU-v in a repeated game is represented as the discounted sum of immediate utilities from each stage; that is,

$$U_v(\infty) = U_v + \delta U_v + \delta^2 U_v + \cdots = \sum_{k=1}^{\infty} \delta^{k-1} U_v, \tag{13}$$

where δ $(0 < \delta < 1)$ is the discount factor which measures how much the SUs value the future utility over the current utility. The larger the value is, the more patient the SUs are. In general, δ is close to 1 for cooperative spectrum sharing in a repeated game. For finite K-$(K > 1)$ stages repeated game, (13) can be rewritten as

$$U_v(K) = \sum_{k=1}^{K} \delta^{k-1} U_v. \tag{14}$$

Without loss of generality, consider a repeated game with two SUs (one is CH and the other is cluster-member) competing for the limited spectrum resources. If an SU senses the PU at the licensed spectrum, it moves to another spectrum hole or stays in the same band without interfering with the PU by adapting its communication parameters such as transmission power or modulation scheme. For this paper, the total transmission power is constrained for SUs to make them stay in the same band during the spectrum sharing.

Figure 3 illustrates the utility region of a repeated game with two SUs for the Gaussian interference channel. U_1 and U_2 are utility functions of user 1 and 2, respectively. Point B can represent that both users transmit with very high power levels and suffer from severe interference, point C or D represents that one user transmits with high power, while the other one uses low transmission power, and point A represents that the two users cooperate by transmitting with appropriate power levels to alleviate interference and improve utility. If the game is only played for only one stage, the NE will correspond to point B, and thus is very inefficient; however, if the game is played for multiple stages, Pareto optimality point A can be achievable, according to the folk theorems.

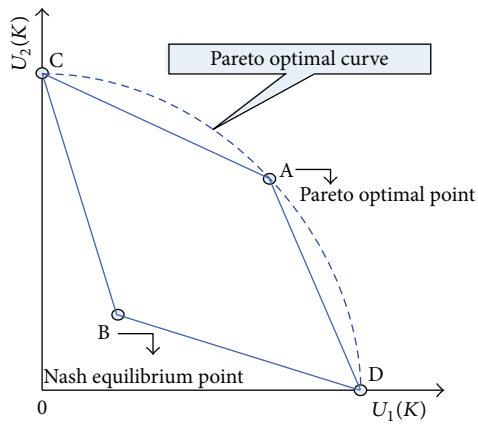


Figure 3: The feasible utility region of a repeated two-player game.

3. Performance Evaluation

The cognitive ad hoc network considered consists of multiple SUs is illustrated in Figure 1. In the figure, TDMA scheme is suggested for every cluster; that is, the TDMA frame is divided into M time slots, and every cluster is assigned one unique spectrum-sharing slot in a frame. The mth cluster always transmits in slots assigned to it.

For a cluster containing only one SU, the SU might occupy the assigned slots for transmission. However, if two or more SUs exist in a cluster, the repeated game should be executed to compete for the assigned slot. Take SU-1 and SU-2 in Figure 1 as an example; they consist of a set of two transmitting-receiving (T-R) pairs with channel gain of 1 for each T-R pair. Assume that the sharing channels, width is normalized to 1, which is divided into two independent channels with width of 1/2 and with the external noise power of N_0. It is noticeable that the transmission powers for SU-1 and SU-2 are P_1 and P_2, respectively, and the total power constraint P for two users is constant with respect to threshold interference power of interference temperature mechanism for CRs.

3.1. Rate Utilities and Total Rate Revenue. When the two users transmit over the same channel, the interference is looked as the Gaussian noise, and the Gaussian interference game was defined in [24, 25].

The interference measured at the receiver u associated with transmitter v is shown to be [26]

$$I_{v,u} = \sum_{u=1,u\neq v}^{V} gP_v f\left(s_v, s_u\right), \tag{15}$$

where P_v denotes the set of transmission power associated with user v over the transmission channel and g the interference gain with symmetric and identical channel conditions. $f(s_v, s_u)$ is the interference function characterizing the interference caused by SU-v to SU-u and is defined as

$$f\left(s_v, s_u\right) = \begin{cases} 1 & \text{if transmitters } v \text{ and } u \text{ are transmitting} \\ & \text{over the same channel,} \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

It is apparent that utilities of SUs are closely related to interferences. Moreover, the performance of the spectrum-sharing algorithm depends significantly on the choice of the utility function which characterizes the preference of a user for a particular channel. The choice of a utility function is not unique. It must be selected to have physical meaning for the particular application and also to have appealing mathematical properties that guarantee equilibrium convergence for the game process. Our objective is to maximize the total rate revenue of SUs by cooperatively sharing the spectrum. Let U_1 and U_2 be the utility functions of SU-1 and SU-2 for each stage of the repeated game, respectively, according to Shannon theory, which can be obtained by

$$\begin{aligned} U_1 &= R_1 = \frac{1}{2}\log_2\left(1 + \frac{P_1}{N_0}\right), \\ U_2 &= R_2 = \frac{1}{2}\log_2\left(1 + \frac{P_2}{N_0}\right), \end{aligned} \tag{17}$$

where R_1 and R_2 are the transmitting rates available for SU-1 and SU-2 and P_1 and P_2 are the transmission power of the two SUs.

By substituting (17) into (14), $U_1(K)$ and $U_2(K)$ are given by the following equations:

$$\begin{aligned} U_1(K) &= \sum_{k=1}^{K} \delta^{k-1} U_1 = \frac{1}{2}\log_2\left(1 + \frac{P_1}{N_0}\right)\sum_{k=1}^{K}\delta^{k-1}, \\ U_2(K) &= \sum_{k=1}^{K} \delta^{k-1} U_2 = \frac{1}{2}\log_2\left(1 + \frac{P_2}{N_0}\right)\sum_{k=1}^{K}\delta^{k-1}, \end{aligned} \tag{18}$$

where K is the number of stages for repeated game and $U_1(K)$ and $U_2(K)$ are the rate utilities of SU-1 and SU-2 for repeated game spectrum sharing.

Then the total rate revenue is

$$\begin{aligned} U(K) &= U_1(K) + U_2(K) \\ &= \frac{1}{2}\log_2\left(1 + \frac{P_1 + P_2}{N_0} + \frac{P_1 P_2}{N_0^2}\right)\sum_{k=1}^{K}\delta^{k-1}. \end{aligned} \tag{19}$$

3.2. Convergence Analysis. As mentioned, revenue stability is closely related to the number of stages. The definition is following: when the revenue differences $\Delta U(k)$ between k-stages repeated game, and $(k-1)$-stages repeated game is less than convergence coefficient ε ($\varepsilon \geq 0$) that is, $\Delta U(k)$ satisfies;

$$0 \leq \Delta U(k) \leq \varepsilon. \tag{20}$$

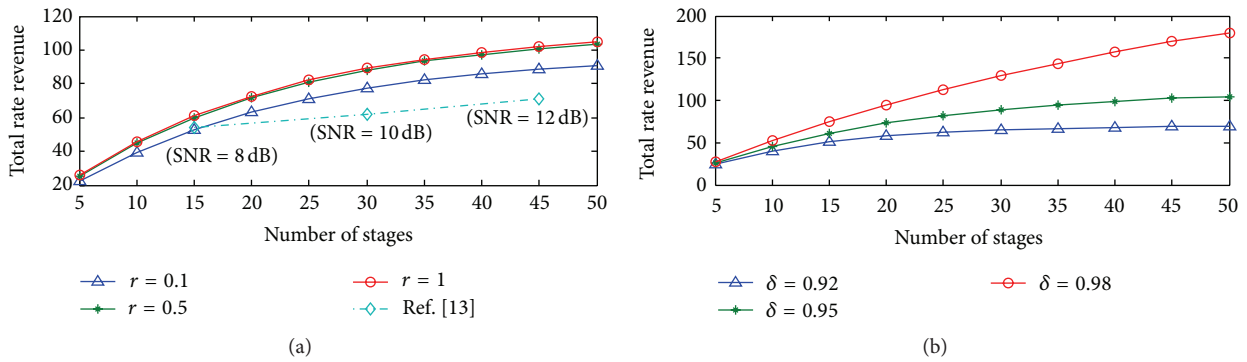


FIGURE 4: (a) Total rate revenue compared with [13]. (b) Total rate revenue versus number of stages for different discount factors.

It is believed that the spectrum sharing achieves convergence. According to (19), $\Delta U(K)$ is shown to be

$$\Delta U(k) = \delta^{k-1}(U_1 + U_2) = \frac{1}{2}\delta^{k-1}\log_2\left(1 + \frac{1}{N_0} + \frac{P_1P_2}{N_0^2}\right). \quad (21)$$

Then, the convergent condition of the algorithm is written as

$$0 \le \frac{1}{2}\delta^{k-1}\log_2\left(1 + \frac{1}{N_0} + \frac{P_1P_2}{N_0^2}\right) \le \varepsilon. \quad (22)$$

The convergent condition is satisfied, if and only if

$$k \ge \log_\delta\left(\frac{2\varepsilon}{\log_2(1 + 1/N_0 + P_1P_2/N_0^2)}\right) + 1. \quad (23)$$

Let K_{cvg} denote the number of stages for convergence; that is,

$$K_{cvg} = \left\lfloor \log_\delta\left(\frac{2\varepsilon}{\log_2(1 + 1/N_0 + P_1P_2/N_0^2)}\right) + 1 \right\rfloor. \quad (24)$$

3.3. Fairness Analysis and Improvements. To maintain reliable communication, a certain transmitting rate threshold required for SUs is necessary. If one user transmits with high power while the other uses low transmission power (as point C or D in Figure 3), the normal communication is not well guaranteed.

How to improve the fairness of spectrum sharing is an urgent issue to be settled. It is assumed that U_1 or U_2 is smaller than the given rate threshold, R_{min}, at the beginning. The rate utilities U_1 and U_2 can be changed by adjusting the transmission powers. The power adjustment procedure for SU-i is executed as follows.

Let the fixed-step size of power adjustment, ΔP, be

$$\Delta P = \left|\frac{P_{ini} - P_{min}}{m}\right|, \quad (25)$$

where P_{ini} is the initial transmission power of the user, P_{min} is the minimum power required correspondence with the rate threshold R_{min} for reliable communication, and m is the number of times for power adjustment. The transmission power for the kth game can be expressed by

$$P_k = P_{k-1} + \Delta P, \quad (26)$$

where P_{k-1} is the transmission power of the $(k-1)$th game.

4. Simulation Results

It is assumed that the noise power $N_0 = 0.01$ W and the total power constraint $P = P_1 + P_2 = 1$ W. Figure 4(a) shows the impact of the number of stages K on the total rate revenue with varying transmission powers for $\delta = 0.95$. When the transmission powers for SU-1 and SU-2 are equal (i.e., ratio of transmission powers $r = P_1 : P_2 = 1$), the total revenue reaches the maximum. It can be seen that, as expected, with the number of stages K increasing, the total revenue increases. It is noticeable that the total revenue changes slowly when K is more than 30. This is because the revenue is tending towards stable condition. Compared with the spectrum allocation algorithm in [13], it is shown clearly that our spectrum-sharing strategy outperforms the algorithm in [13], and the algorithm in [13] only performs well at high SNR.

The total rate revenue versus different number of stages with varying discount factors for $r = 1$ is plotted in Figure 4(b). We can see that the total revenue increases with δ increasing.

Figure 5(a) shows the impact of transmission power on the convergence rate when the noise power $N_0 = 0.01$ W, discount factor $\delta = 0.95$, and convergence coefficient $\varepsilon = 0.1$. The number of stages for convergence K_{cvg} increases when the difference between P_1 and P_2 is large enough (i.e., $r < 0.4$). Nevertheless, with r increasing, the number of stages for convergence almost remains unchanged. In this case, the total revenue can reach the maximum (as shown in Figure 4).

Figure 5(b) plots the convergence behavior with varying discount factor δ for noise power $N_0 = 0.01$ W, ratio of transmission powers $r = 1$, and convergence coefficient $\varepsilon = 0.1$. There is a significant increase in the number of stages for convergence K_{cvg} with δ increasing. This is because the larger the value δ is, the more patient the players are, and the convergence rate becomes more slowly. Specifically, K_{cvg} is an exponential function of δ when the value δ is more than 0.96.

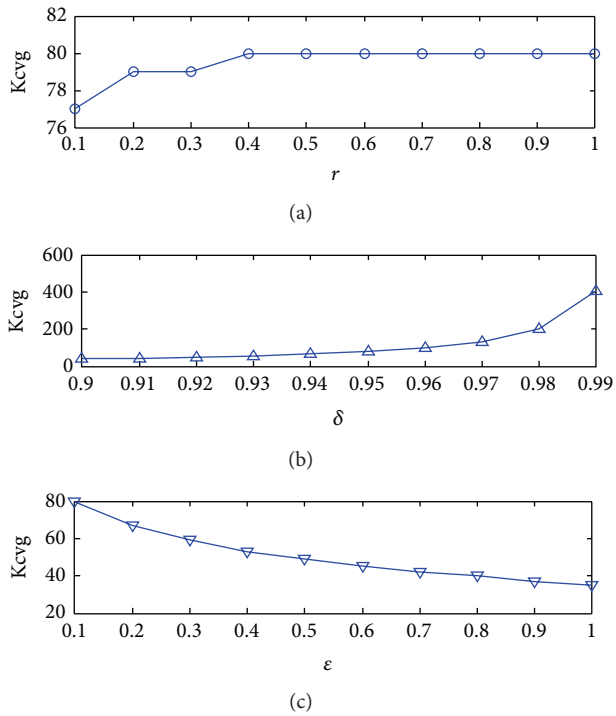


Figure 5: (a) Convergence performance versus different transmission powers. (b) Convergence performance versus discount factor. (c) Convergence performance versus convergence coefficient.

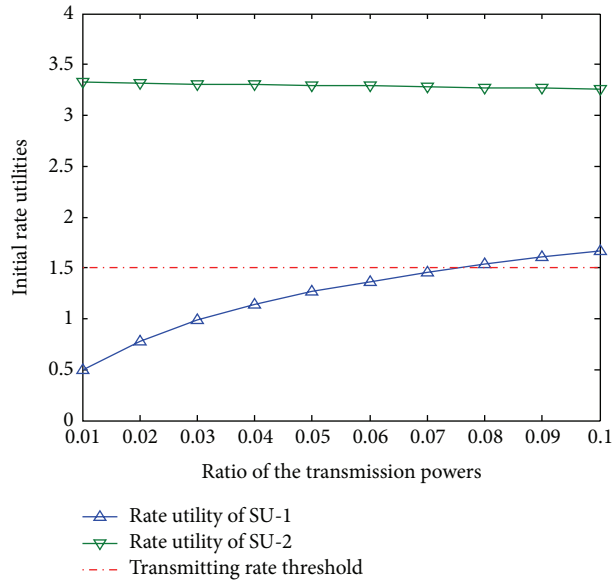


Figure 6: Initial rate utilities for different transmission powers.

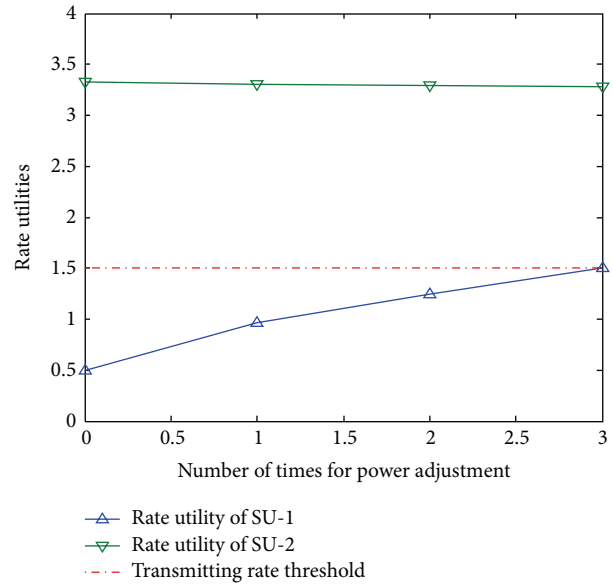


Figure 7: Rate utilities adaptation to power adjustment.

Figure 5(c) depicts the number of stages for convergence K_{cvg} in terms of convergence coefficient ε, for noise power $N_0 = 0.01$ W, ratio of transmission powers $r = 1$ and discount factor $\delta = 0.95$. It can be seen that the smaller value ε is, that is, more stringent convergence condition, the more slowly convergence rate is.

The rate utilities U_1 and U_2 for different transmission powers are presented in Figure 6. It is observed that when the transmission power difference between the two users is large at the beginning of a repeated game, that is, U_1 or U_2 is smaller than the given rate threshold (a predefined value), then the reliable communication is not to be guaranteed. Spectrum sharing is only one side for this case, and the fairness of SUs is nothing to speak of. However, with the transmission power difference decreasing, the fairness is improved.

The adaptation of rate utility due to the transmission power adjustment is shown in Figure 7. As expected, when the rate utility U_1 is smaller than the given threshold, U_1 can be changeable for meeting the requirement of transmitting rate with very little rate utility loss of U_2. Thus, the fairness of the algorithm is well guaranteed.

The total revenue comparison between the proposed algorithm with the method (LAG) in [16] is plotted in Figure 8. It can be observed that the total rate revenue increases with better fairness (i.e., larger ratio of the transmission powers, r). Especially, the total revenue is maximum when $r = 1$. It is also noted from the figure that when the access probability P is less than a value, that is, $P \leq 0.7$ the obtained total rate revenue of the proposed algorithm outperforms the LAG algorithm. As the access probability increases, that is, $P > 0.7$, there is an increasing revenue gap for LAG algorithm. However, as mentioned in [17], the collision happened as long as L ($L \geq 2$) SUs in the transmission range of each other has data to transmit simultaneously. That is, larger access probability is hardly guaranteed in CRN.

5. Conclusions

In this paper, the spectrum sharing is modeled as a repeated game under which cluster-member SU communicates with only cluster-head SU, and frequent collisions between SUs

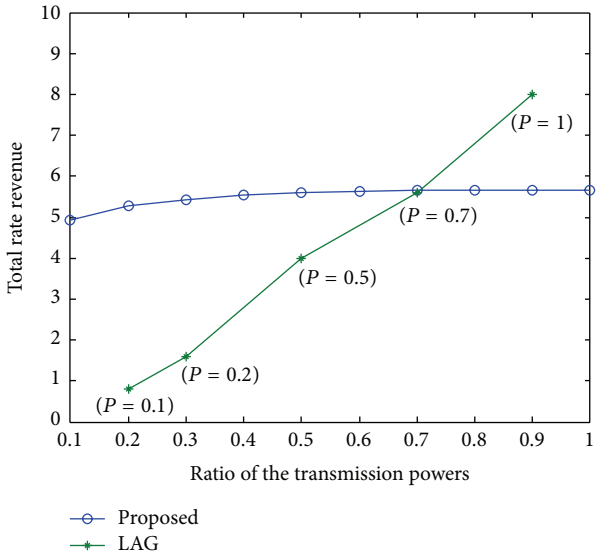


Figure 8: Total rate revenue compared with LAG.

are avoided than that under no-cluster policy. The aim of this work is to maximize the total rate revenue of SUs under the repeated game model, while meeting the fairness requirements for transmitting data. The first step toward this work is to analyze convergence condition under which the total rate revenue of SUs is maximized. The analysis shows that the transmission powers, discount factor, and convergence coefficient affect the total rate revenue. The analysis continues by considering the fairness of spectrum sharing, and the fairness is improved by adjusting the transmission powers. Simulation results demonstrate that the proposed spectrum-sharing algorithm can achieve better performance than the preexisting ones in terms of the total rate revenue and fairness of spectrum sharing.

For future research, the QoS (Quality-of-Service) requirement for SUs will be considered for spectrum sharing.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (61261014), the State Key Laboratory of Rail Traffic Control and Safety (RCS2011K006), the Beijing Jiaotong University, and the Fundamental Research Funds for the Gansu Universities (212087-2).

References

[1] J. Mitola and G. Q. Maguire, "Cognitive radio: making software radios more personal," *IEEE Personal Communications*, vol. 6, no. 4, pp. 13–18, 1999.

[2] S. Haykin, "Cognitive radio: brain-empowered wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 2, pp. 201–220, 2005.

[3] I. F. Akyildiz, W. Y. Lee, M. C. Vuran, and S. Mohanty, "Next generation/dynamic spectrum access/cognitive radio wireless networks: a survey," *Computer Networks*, vol. 50, no. 13, pp. 2127–2159, 2006.

[4] R. Cendrillon, W. Yu, M. Moonen, J. Verlinden, and T. Bostoen, "Optimal multiuser spectrum balancing for digital subscriber lines," *IEEE Transactions on Communications*, vol. 54, no. 5, pp. 922–933, 2006.

[5] J. Huang, R. A. Berry, and M. L. Honig, "Spectrum sharing with distributed interference compensation," in *Proceedings of the 1st IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (IEEE DySPAN '05)*, pp. 88–93, Baltimore, Md, USA, November 2005.

[6] H. A. Bany Salameh, M. Krunz, and O. Younis, "Cooperative adaptive spectrum sharing in cognitive radio networks," *IEEE/ACM Transactions on Networking*, vol. 18, no. 4, pp. 1181–1194, 2010.

[7] D. Niyato and E. Hossain, "Competitive spectrum sharing in cognitive radio networks: a dynamic game approach," *IEEE Transactions on Wireless Communications*, vol. 7, no. 7, pp. 2651–2660, 2008.

[8] R. Etkin, A. Parekh, and D. Tse, "Spectrum sharing for unlicensed bands," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 3, pp. 517–528, 2007.

[9] B. Wang, Z. Ji, and K. J. R. Liu, "Self-learning repeated game framework for distributed primary-prioritized dynamic spectrum access," in *Proceedings of the 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '07)*, pp. 631–638, San Diego, Calif, USA, June 2007.

[10] C. K. Tan, M. L. Sim, and T. C. Chuah, "Game theoretic approach for channel assignment and power control with no-internal-regret learning in wireless ad hoc networks," *IET Communications*, vol. 2, no. 9, pp. 1159–1169, 2008.

[11] Z. Han, C. Pandana, and K. J. K. Liu, "Distributive opportunistic spectrum access for cognitive radio using correlated equilibrium and no-regret learning," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '07)*, pp. 11–15, Hong Kong, March 2007.

[12] B. B. Wang, Y. Wu, and Z. Ji, "Game theoretical mechanism design methods," *IEEE Signal Processing Magazine*, vol. 25, no. 6, pp. 74–84, 2008.

[13] M. Liang and Q. Zhu, "A new algorithm of spectrum allocation for cognitive radio based on cooperative game," in *Proceedings of the 6th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '10)*, Chengdu, China, September 2010.

[14] A. Montanari and A. Saberi, "Convergence to equilibrium in local interaction games," in *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS '09)*, pp. 303–312, Atlanta, Ga, USA, October 2009.

[15] H. Li and Z. Han, "Competitive spectrum access in cognitive radio networks: graphical game and learing," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '10)*, pp. 1–6, Sydney, Australia, April 2010.

[16] Y. H. Xu, J. L. Wang, Q. H. Wu, A. Anpalagan, and Y. D. Yao, "Opportunistic spectrum access in cognitive radio networks-global optimization using local interaction games," *IEEE Journal of Selected Topics in Signal Processing*, vol. 6, no. 2, pp. 180–194, 2012.

[17] C. Zou and C. Chigan, "A game theoretic DSA-driven MAC framework for cognitive radio networks," in *Proceedings of the IEEE International Conference on Communications (ICC '08)*, pp. 4165–4169, Beiing, China, May 2008.

[18] M. J. Osborne, *An Introduction to Game Theory*, Oxford University Press, 2004.

[19] M. Chatterjee, S. K. Das, and D. Turgut, "WCA: a weighted clustering algorithm for mobile Ad Hoc networks," *Journal of Clustering Computing*, vol. 5, no. 2, pp. 193–204, 2002.

[20] W. C. Y. Lee, *Mobile Cellular Telecommunications*, McGraw Hill, 1995.

[21] G. X. Jiang and Z. Y. Yang, "A distributed clustering algorithm based on δ-Cluster Stability for Mobile Ad hoc Networks," in *Proceedings of the 4th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '08)*, pp. 1–6, Dalian, China, October 2008.

[22] X. Du, D. Wu, W. Liu, and Y. Fang, "Multiclass routing and medium access control for heterogeneous mobile ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 55, no. 1, pp. 270–277, 2006.

[23] Z. Ji and K. J. R. Liu, "Cognitive radios for dynamic spectrum access—dynamic spectrum sharing: a game theoretical overview," *IEEE Communications Magazine*, vol. 45, no. 5, pp. 88–94, 2007.

[24] W. Yu, G. Ginis, and J. M. Cioffi, "Distributed multiuser power control for digital subscriber lines," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 5, pp. 1105–1115, 2002.

[25] A. Laufer and A. Leshem, "Distributed coordination of spectrum and the prisoner's dilemma," in *Proceedings of the 1st IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySpan '05)*, pp. 94–100, Baltimore, Md, USA, November 2005.

[26] N. Nie and C. Comaniciu, "Adaptive channel allocation spectrum etiquette for cognitive radio networks," in *Proceedings of the 1st IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN '05)*, pp. 269–278, November 2005.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 301869, 10 pages
http://dx.doi.org/10.1155/2013/301869

Research Article

Clustering with One-Time Setup for Reduced Energy Consumption and Prolonged Lifetime in Wireless Sensor Networks

Heewook Shin, Sangman Moh, and Ilyong Chung

Department of Computer Engineering, Chosun University, 309 Pilmundaero, Dong-gu, Gwangju 501-759, Republic of Korea

Correspondence should be addressed to Sangman Moh; smmoh@chosun.ac.kr

Received 17 October 2012; Accepted 22 January 2013

Academic Editor: Wei Wang

In wireless sensor networks, clustering is effectively used for many applications, including environment monitoring, because it promises efficient energy consumption for inexpensive battery-operated sensors. The most famous clustering protocol, LEACH (Low Energy Adaptive Clustering Hierarchy), enables the balanced consumption of energy to prolong a network lifetime. In LEACH, however, extra energy and time are consumed to reform clusters at the setup phase of every round. This side effect is worse as the number of clusters increases. This paper presents a novel energy-efficient clustering scheme called COTS (Clustering with One-Time Setup) which removes the cluster-reforming process required at every round after the first round. The proposed COTS allows that the role of the cluster head is rotated among members in a cluster without cluster reforming. By removing the cluster-reforming process, the number of transmissions per round is decreased accordingly. As a result, energy consumption is significantly reduced, resulting in prolonged network lifetime. The simulation study shows that the network performance and lifetime are much improved as the number of clusters is increased.

1. Introduction

Wireless sensor networks (WSNs) consist of many battery-powered sensor nodes (SNs) that monitor their physical surroundings and send the resulting data to a sink node. Since the battery resource directly affects the operation time of sensors, it is very important in prolonging the lifetime of a WSN to design energy-efficient protocols. Thus, many studies in WSNs have focused on delivering the sensed data to the destination while being energy efficient.

Routing in WSNs means that the information from SNs is forwarded to the base station (BS) regularly or on demand. There are two types of routing, classified as flat and hierarchical routing. A clustering approach can be regarded as a hierarchical routing technique. As reported in many studies, clustering schemes can save a lot of energy in WSNs [1, 2]. The clustering associated with data aggregation improves network performance by decreasing the amount of data to be delivered and the number of hops from sensors to the BS. In such networks, however, more energy is consumed in the cluster head (CH) nodes because more computing and communication loads are assigned to the CHs. This non-uniformity of energy consumption among nodes results in some nodes dying earlier than others.

LEACH (Low Energy Adaptive Clustering Hierarchy) is the most famous clustering protocol that resolves the energy unbalancing problem among nodes [3]. In LEACH, nodes are classified into two groups: CHs and SNs. The main idea of LEACH is to reform clusters once every period of time, called a round, in order to rotate the role of the CH among members in a cluster. There have been many studies in the past ten years exploring the LEACH protocol to improve performance. However, there has been no work to address energy consumption during the cluster-reforming process. Since LEACH was developed, many works have been reported. LEACH-C (LEACH Centralized) [4] is one of LEACH's variations, in which cluster heads are elected by a base station to prevent energy imbalance. In HEED (Hybrid, Energy-Efficient Distributed) clustering [5], residual node energy is taken into consideration for the dispersion of energy

consumption. In BCDCP (Base station Controlled Dynamic Clustering Protocol) [6], cluster heads are selected from a set of candidate nodes. In TEEN (Threshold sensitive Energy-Efficient sensor Network protocol) [7], a threshold value is set on the basis of LEACH to reduce the energy consumption of CHs and sensor nodes. In APTEEN (Adaptive TEEN) [8] which is a combination of LEACH and TEEN, a time period is set for transmitting data periodically for data accuracy and reliability.

In this paper, we take repetitive cluster reforming over the network lifetime into account for reduced energy consumption and prolonged lifetime in WSNs. During the setup phase of every round in LEACH, the cluster reforming process is carried out by all the nodes in a cluster. That is, every SN transmits at least once, either to inform others that it is a CH or to request to join a chosen cluster. Furthermore, the CH broadcasts the TDMA schedule to all members. Notice again that the setup phase, including communications, is performed at every round in the conventional LEACH protocol.

In this paper, we propose an energy-efficient clustering scheme called COTS (Clustering with One-Time Setup) by removing the cluster-reforming process and adding a rescheduling slot to the end of every round. Usually, the setup phase is composed of hundreds of slots, even though it depends on the number of nodes and the pattern of random access. By skipping this cluster-reforming process, energy is significantly saved. In COTS, the role of the CH is rotated among the members in a cluster by transmitting the cluster head order at the rescheduling slot. As a result, energy consumption is significantly reduced, and the network lifetime is increased accordingly. Our simulation study shows that the proposed COTS remarkably improves network performance and lifetime.

The main idea of the proposed COTS can be summarized in the following steps.

(i) A cluster is formed just one time at the setup phase of the first round.

(ii) Once a cluster is formed, the CH creates a cluster head list. This list consists of all the other member nodes in order of the closest to the furthest away. This list is used to rotate the role of CH among all other member nodes. The list is broadcasted to all the other members during the setup phase at the first round.

(iii) In the steady-state phase, the CH collects data from members as in other protocols. If a CH cannot receive data from a member, the CH regards the member as a dead node.

(iv) At the rescheduling slot every round, the CH creates a new cluster head order based on the last set of collected data packets and then broadcasts the new cluster head order to members. If a member does not receive the order, it simply invalidates the current cluster head order.

The rest of this paper is organized as follows. In the following section, the LEACH protocol's strengths and weaknesses are reviewed in brief. Section 3 presents the proposed COTS algorithm with respect to design principles and operations. The performance of the proposed COTS is evaluated via simulation and compared to the conventional protocol in Section 4. Finally, this paper is concluded in Section 5.

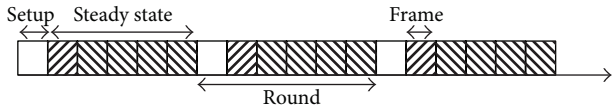


Figure 1: Operation of the LEACH protocol.

2. Preliminaries

A lot of studies on clustering-based protocols in WSNs have been published. LEACH [3] is one of the most famous clustering techniques. It divides an entire network into groups called clusters. A cluster consists of many SNs and a CH. The CH collects sensed data from SNs and then aggregates and transmits them to the BS. Hence, SNs do not need to communicate with the base station, resulting in a decreased communication burden. Therefore, energy consumption is reduced and network lifetime increased.

In hierarchical approaches such as clustering, the data at the higher levels of hierarchy is more important than those at the lower levels. Thus, communications between CH and BS need more attention compared to intracluster communications, because packet loss between CH and BS means a loss of all the data in a cluster. Therefore, the CH-to-BS signal uses a Carrier Sensing Multiple Access (CSMA) technique, and it is broadcasted to the entire network to avoid the hidden terminal problem. On the other hand, the sensor-to-CH signal uses Time Division Multiple Access (TDMA).

The LEACH protocol repeats a series of setup and steady-state operations with the static time interval called round. As shown in Figure 1, each round consists of two phases: a setup phase and a steady-state phase.

2.1. Setup Phase. In the setup phase, clusters are formed, and the TDMA schedule is created for the steady-state phase. Every node wakes up and initializes its internal state as default. Then each node generates a random number between 0 and 1 and compares the number with the $T(n)$ value. If it is smaller than $T(n)$, the node will be a new CH; otherwise, it is a member of a cluster. $T(n)$ can be defined as

$$T(n) = \begin{cases} \dfrac{p}{1 - p\left(r \bmod 1/p\right)} & \text{if } n \in G \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

where p is the desired percentage of CHs over the total number of nodes, r is the identifier of the current round, and G is the set of nodes that have not clustered in the last $1/p$ rounds [3, 9].

After the election of CHs, all the nodes in a network perform the operation depicted in Figure 2. The left-hand side of Figure 2 represents the behaviors of CHs, while the

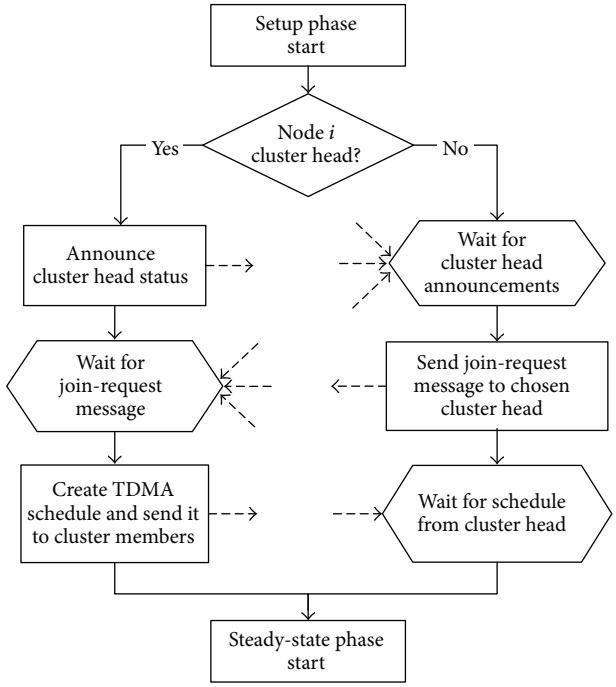


Figure 2: The setup phase in LEACH.

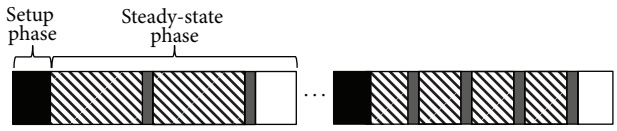


Figure 3: Rounds of the LEACH protocol.

right-hand side represents those of cluster members. The elected CHs broadcast their existence to all the nodes of the network. SNs receive these advertisements and choose one of the nearest CHs as their CH. An SN sends the cluster join message to the selected CH. Hence, CHs can know all of their members. Each CH creates a TDMA schedule for the steady-state phase and broadcasts the schedule to its members. Finally, cluster members receive the schedule. During the setup phase, every member node sends the Join-Request message to its CH as shown in Figure 2. This causes long delay because a CH has to receive Join-Request messages from its member nodes in a serialized fashion. In COTS, however, the long delay is permanently removed after the first round because there is only one-time setup in COTS, which will be presented in Section 3.

2.2. Steady-State Phase. In the steady-state phase, cluster members sense the surroundings and transmit the sensed data to their CH depending on the TDMA schedule received at the setup phase. SNs go into sleep mode to save energy for other slots. As shown in Figure 3, a steady-state phase consists of a few frames, and a frame can be divided in two time slots: the time slot for the SNs and the time slot for CH. SNs transmit sensed data to their CH in the time slot for CH. The CH compresses (or aggregates) this data and transmits it to the BS. Since a cluster operates in a frame unit, if it does not have enough time for a frame, the cluster will not work in the time left.

2.3. Weaknesses of the LEACH Protocol. Figure 3 shows the rounds of the LEACH protocol. Black squares indicate the setup phase, squares with diagonal lines are multiple TDMA time slots for sensors, and dark gray squares represent the time taken by CHs to compress data and transmit it to the BS. The left side of the dotted line is the initial round, and the right side represents the final round at the end of the network lifetime. It shows a LEACH weakness that the transmission interval depends on the number of members in a cluster [10]. For example, we assume that the time of a steady-state phase is 5 seconds, and all time slots spend 0.1 seconds equally. If the number of members is 10, each SN has 5 transmission time slots. On the other hand, if the number of members is 5, each SN can occupy 10 slots. The fewer the number of alive nodes, the greater the number of detections will happen for a member.

If the detection distance for sensing is not changed, it does not have any advantage to increase the number of detections because decreasing the sensing interval of nodes causes more energy consumption. Thus, creating a TDMA schedule based on the number of living nodes can be disadvantageous in terms of energy efficiency.

Another weakness is caused by the repetitive election of CHs based on the changed probability of the total number of living nodes. Hence, there are differences in the number of clusters between rounds. An increment of the number of CHs decreases total network lifetime, and a reckless decrease causes unbalanced energy consumption between nodes in the entire network.

3. Clustering with One-Time Setup

In this section, we present the operational principles of the proposed COTS. Clusters are formed at the setup phase of the first round, and the first round only. This results in significant energy savings.

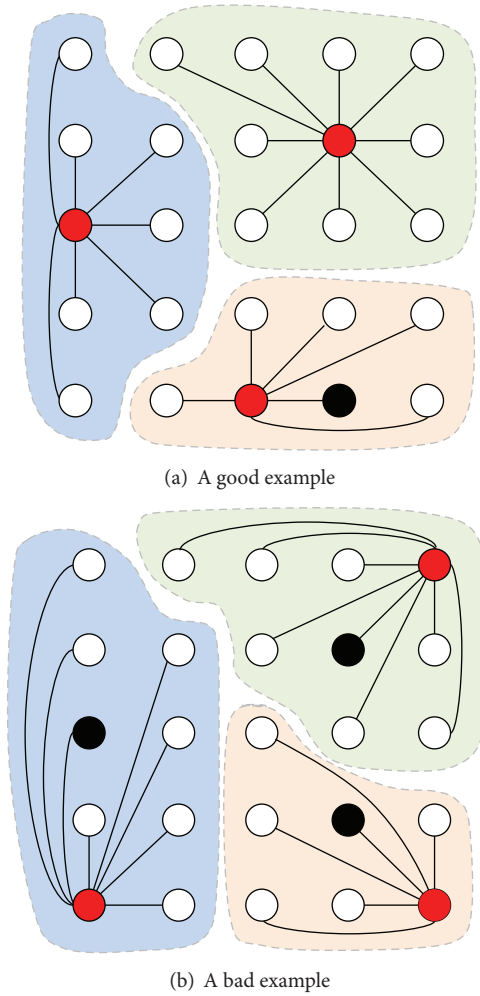
Unlike the conventional approaches such as LEACH, the proposed COTS does not require cluster reforming, and thus the cluster membership is not changed over rounds. The CH order is determined at the setup phase of the first round. Every member of a cluster simultaneously changes its CH according to the CH order in the next round. This practice may run into a problem as the number of dead nodes increases over time. That is, if a dead node is to be a CH at the next round, it results in the loss of all packets of a cluster during that round. There is no such problem in the conventional protocols, because clusters are newly formed at every setup phase. In order to avoid this phenomenon, a new or updated cluster head order should be created and broadcasted when nodes are dead. Thus, we introduce a new time slot for rescheduling CHs called a rescheduling slot, which is positioned just before the new round starts.

3.1. One-Time Setup. After all data is received from member nodes, each CH compresses the aggregated data into a single message and then transmits it to the BS. In LEACH, however, this method has a weakness in that the CH cannot send the message to the BS when the CH does not receive any data from members within the steady-state phase. In the actual code of the MIT *uAMPS* project [3], if CH did not receive the last node's data, it does not compress and transmit data for the BS. This is one of the reasons for LEACH to carry out reclustering per round. However, the proposed COTS does not have a setup phase after the first round, and the cluster membership and the TDMA schedule are not changed at all. So, the TDMA schedule can include unavailable time slots that have been assigned to dead nodes. This is trivial when compared to the significant reduction of energy consumption by removing the setup phase and does not affect performance at all.

3.2. Clustering. COTS is subject to a network topology whose nodes are distributed evenly in the same way as the LEACH protocol. Once clusters are formed after the setup phase, the cluster member is not changed for a network lifetime, even if there are high density areas of nodes that encourage inefficient energy consumption of the specific area. If nodes are distributed uniformly, the network has a lower risk of node concentration.

Figures 4(a) and 4(b) show the cluster state immediately after the setup phase in the network using a LEACH protocol on nodes that are distributed evenly. White points represent SNs, red points are CHs, and black points indicate the best CH position required by each cluster. The best CH position means the CH location that minimizes the total communications cost. A transmission range is a critical factor that affects the energy consumption of nodes. It should be noted that almost all CHs are located in a relatively good position in LEACH, but are not always as good as in Figure 4(a). In the process of cluster reforming, each node measures the distances between CHs and chooses a cluster made by a CH with the shortest distance. Thus, some nodes have no choice but to select the CH located at the relatively bad position from outside the cluster, if the distances to other CHs are longer than the CH. There is the probability that some nodes located in the corner of a network cluster become CHs. Figure 4(b) shows a network with the CHs located in the corner. One glance is enough to know that the position of the CHs is not good. In this case, the transmission length between nodes of a cluster becomes the longest, resulting in higher energy consumption. On the other hand, the black points represent the ideal CH position with the most efficient energy consumption.

We can explain the features of the network without cluster reforming by using Figure 5, which shows the state of the clusters in Figure 4(a) after a round. There are three different phenomena as contrasted with the LEACH protocol having reclustering, and we explain these as the three clusters drawn in the figure. The blue cluster shows the worst case of CH positions made by rotating the role of CH within the cluster. If the width and height between two adjacent nodes are taken as 1 unit, the total length for communications of the blue cluster of Figure 4(a) is about 8.4 and for Figure 5 is 16.8. The length of Figure 5 is two times as long as Figure 4(a), but this does not indicate the doubled energy consumption. According to the LEACH energy consumption model, the energy consumption increases as the second or fourth power of the distance, so it has a bad effect on network lifetimes. Thus, we need to know the effect of the longer communication length and the effect of not reforming clusters, comparing the worst cases in cluster nonreforming and reforming, because the worst case affects the lifetime of a network more than the best case. The difference of the members between the two blue clusters is 1, and the difference of the total length is also just 1. If the number of nodes of a network increases, these differences will also increase. The more important thing is that the cluster with fixed cluster members for rounds will inevitably have all cases, including the worst case, when the network has enough rounds. Otherwise, the network that reforms has more good cases, because the new elected clusters have new members, which join depending on distance. Nodes (1, 1) and (2, 2) of Figure 5 belong to the blue cluster, although these are closer to the green cluster. Likewise, node (5, 2) also cannot join the blue cluster. Removing cluster reformation removes the chances for some nodes to



(a) A good example

(b) A bad example

Figure 4: Clustering examples of the LEACH protocol.

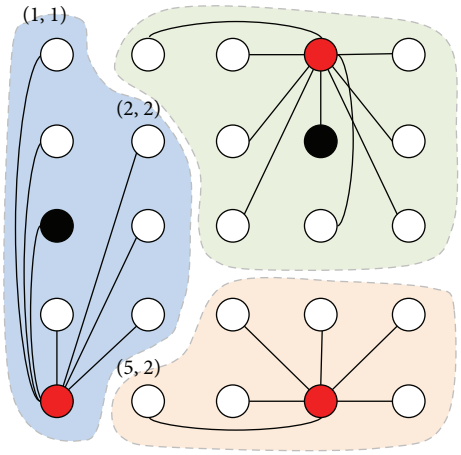


Figure 5: Clustering example of the COTS protocol.

participate in reasonable clusters and therefore increases the total communication length. In other words, the flexibility of clustering is decreased by fixing the members of a cluster. For these reasons, a protocol without cluster reformation needs to minimize the negative effects of fixed clusters. To do so, COTS creates the CH order based on how far each node is from all other nodes in the cluster during the setup phase. Once the setup phase is finished, the CH knows which other node has the next best position to be the next CH. Thus these nodes can be selected as CHs for the remaining rounds to reduce the negative effects of outlying CHs. The red cluster of Figure 5 has the best positioning compared with those of Figure 4. And with a cluster head order determined by better geographical positioning, the flexibility decrease problems are delayed for as long as possible to only appear near the end of a network's lifetime.

3.3. Dynamic Rescheduling. Figure 6 shows the operational procedure of the steady-state phase in the proposed COTS. CH aggregates the sensed data and transmits it to the BS every round. In the meantime, CH saves a list of living member nodes from which sensed data is successfully received. From the list of living members, a new or updated order of CHs is built. This list will remain in effect until a new CH order is created during another round.

The order of CHs is rotated for balanced energy consumption. Figure 7 shows the rounds of the COTS protocol including the rescheduling slot. The proposed COTS includes a rescheduling slot every round. In the rescheduling slot, CH has a chance to announce the living status of cluster members. If dead nodes are detected by a CH, the CH announces a new or updated cluster head order to all of the members within the cluster. The member nodes receive the cluster head order. The cluster head order is updated, when there is any difference between the current cluster head order and a new cluster head order. Otherwise, the CH sends alive messages to members to inform them of its survival when any node is not dead during the current round. Hence, every member can know the status of all of the other members, and no dead node is chosen as CH, as depicted by the light gray

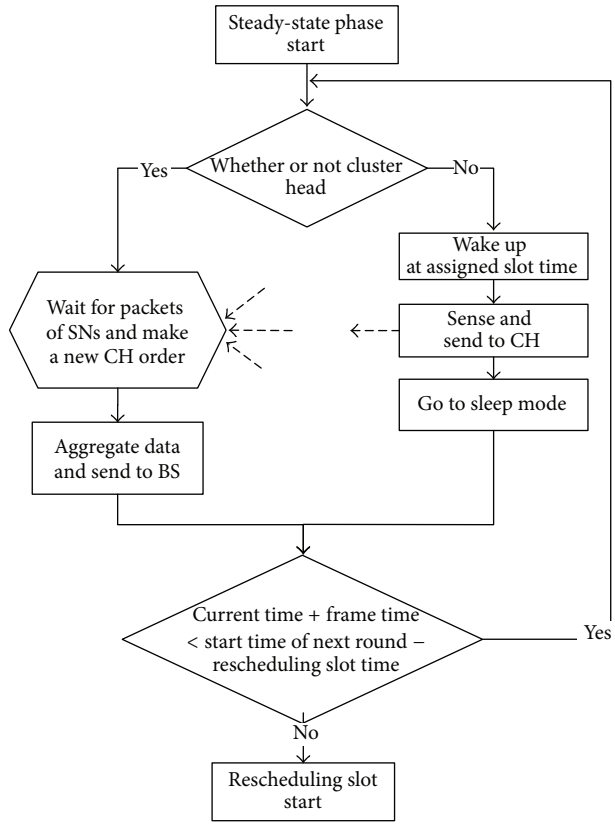


Figure 6: The steady-state phase in COTS.

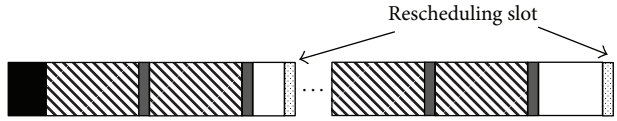


Figure 7: Rounds of the COTS protocol.

squares in Figure 7. Figure 8 shows the node operation at the rescheduling slot. The left-hand side of the first question of the flow represents the operation of the current CH, and the right-hand side represents that of member nodes.

4. Performance Evaluation

In this section, the performance of the proposed COTS is evaluated via extensive simulations using ns-2 [11] and compared with the conventional protocol.

4.1. Simulation Environment. The simulation parameters are shown in Table 1. The initial battery energy assigned to each SN is assumed to be 1 and 2 Joules for different scenarios of energy resource in our simulation. In another simulation, the number of CHs is assigned as 5 and 10 to know what effect the difference of the cluster area has. The ns-2 simulator automatically regulates a limit of transmission of a node described as transmission range by calculating the

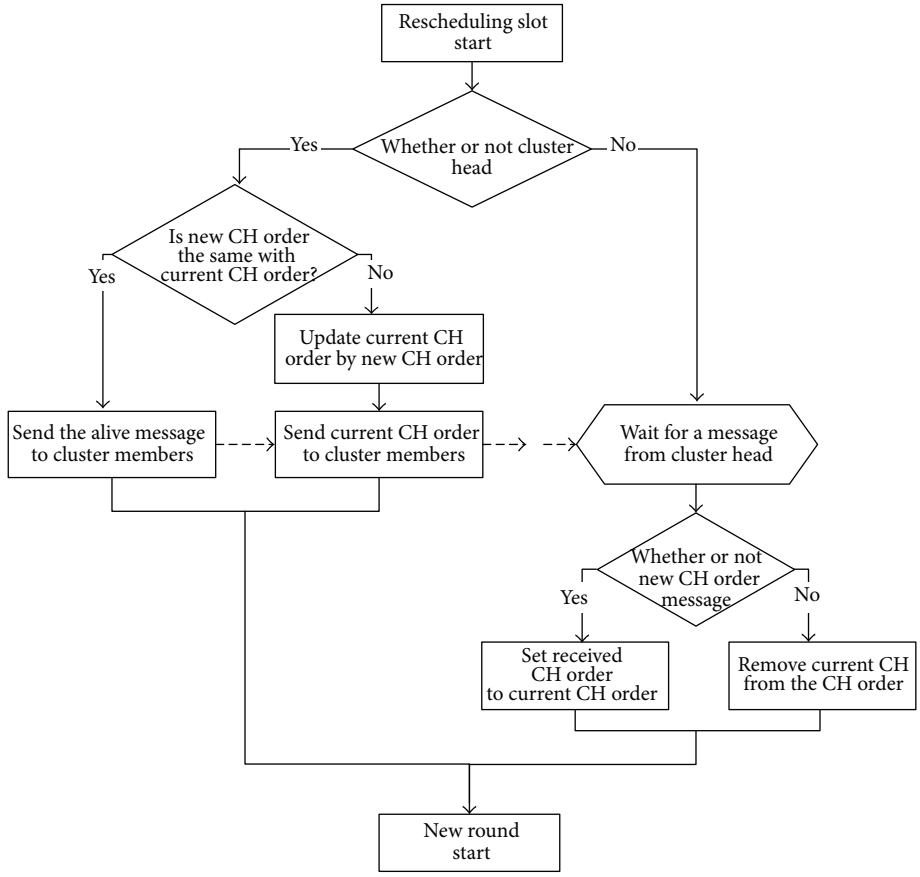


Figure 8: Operation at the rescheduling slot.

Table 1: Parameters for simulation.

Parameter	Value
Number of nodes	100
Number of CHs	5 (default), 10
Initial energy	1, 2 (default)
Round interval	10 sec
E_{da}	5 nJ/bit/signal
E_{elect}	50 nJ/bit
E_{sense}	5 nJ/bit
ε_{fs}	10 pJ/bit/m^2
ε_{mp}	0.0013 pJ/bit/m^4
Location of BS	(125, 75)
Network area	100×100 m^2
Transmission range	136 m

length between base station and the farthest node. It is for minimizing inefficiency in the energy consumption caused by the cluster head advertisement in setup phases. Note here that, as in most WSNs, the transmission power of nodes is fixed, and their communication range is also fixed.

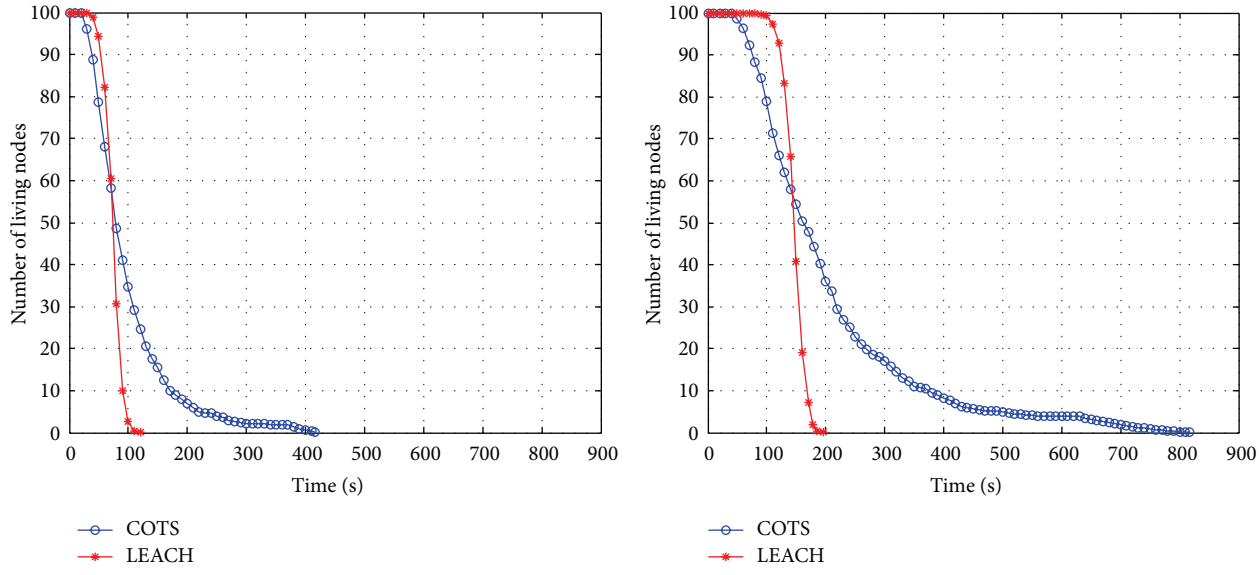
For our experiment, we used the energy consumption model [12, 13] provided with the LEACH source code. The propagation model is the same as that of the LEACH protocol, which does not consider errors in wireless channels. Power control can be used to invert this loss by appropriately setting the power amplifier. That is, if the distance is less than a threshold d_0, the free space (fs) model is used; otherwise, the multipath (mp) model is used [14]. Thus, to transmit a k-bit message along the distance d, radio power consumption is given by

$$E_{Tx}(k,d) = E_{tx-elect}(k) + E_{Tx-amp}(k,d) = \begin{cases} kE_{elect} + k\varepsilon_{fs}d^2, & d < d_0 \\ kE_{elect} + k\varepsilon_{mp}d^4, & \text{otherwise.} \end{cases} \quad (2)$$

The first function of (2) is an energy consumption value spent by the transmission of an electronic device, and the second function is the value by the transmission amplifier. Since receivers do not need to have any amplifiers, it only spends energy for an electronic device as shown in (3). The radio energy consumption for receiving k-bit data is calculated as

$$E_{Rx}(k,d) = E_{Rx-elec}(k) = kE_{elec}, \quad (3)$$

where E_{elec} is the radio electronics transmission/reception energy, which depends on factors such as digital coding, modulation, filtering, and spreading of the signal, $\varepsilon_{fs}{}^2$ and $\varepsilon_{mp}{}^4$ are constant values for the amplifier energy depending



(a) With the initial energy of 1 Joule

(b) With the initial energy of 2 Joules

Figure 9: The number of living nodes.

on the distance to the receiver and acceptable bit-error rate, and E_{da} is the energy consumption of data aggregation. In this paper, the communication energy parameters are set as follows: E_{elec} is 50 nJ/bit, ε_{fs} is 10 pJ/bit/m^2, ε_{mp} is 0.0013 pJ/bit/m^4, and E_{da} is 5 nJ/bit/signal.

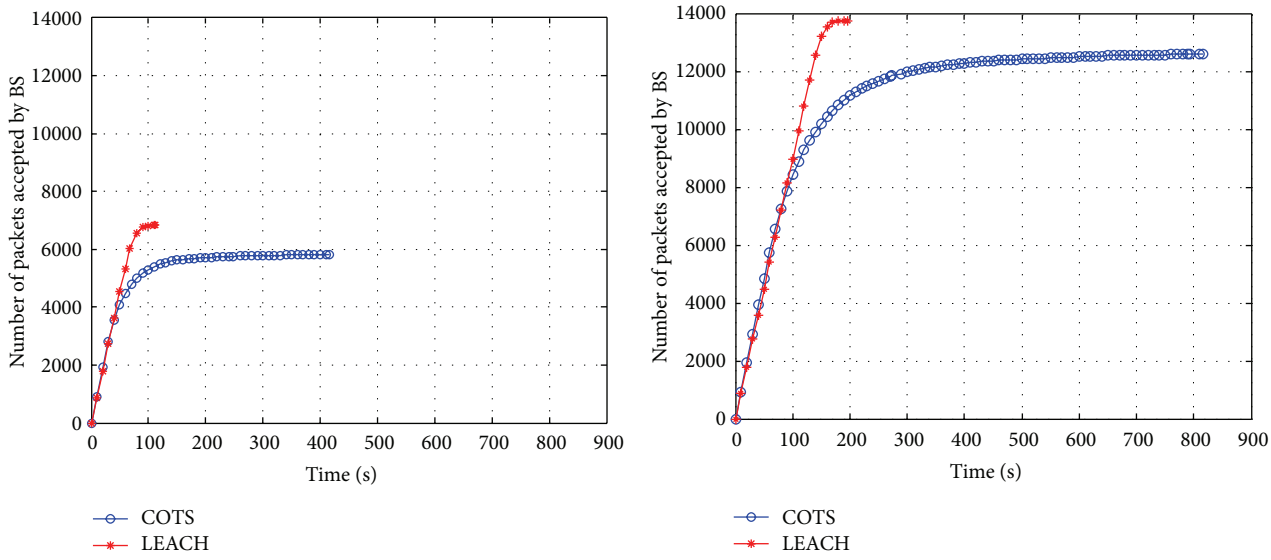
4.2. Simulation Results and Discussion. In WSNs, the ultimate goal of energy saving is to prolong the network lifetime. In other words, "reduced energy consumption" means "prolonged lifetime" in WSNs. The network lifetime is indicated by the number of living sensor nodes. The two graphs of Figure 9 show the network lifetime for the different initial energies of 1 and 2 Joules, respectively. For the two scenarios of energy resources, there is an improvement of 34% and 37%, respectively.

In the graphs, the number of living nodes in COTS is less than that in LEACH until 75 and 150 seconds for initial energies of 1 and 2 Joules, respectively. This indicates that the energy is consumed faster. This is due to the beneficial improvement in COTS, which is as follows. Given a period of time, more rounds are carried out in COTS compared to LEACH, because the setup phase is removed every round after the first round in COTS. More rounds mean both more sensing and more transmissions, resulting in more energy consumption during the same period of time. Conceptually, COTS replaces the setup phase by just one time slot, called the rescheduling slot. As a result, the number of frames in COTS is more than that in LEACH. So, both the configuration time and the energy are remarkably decreased.

The two graphs of Figure 10 represent the number of packets accepted by BS. Note that the curves of LEACH and COTS end at the network lifetime of them, respectively, because no packets are accepted by BS after the lifetime. At the beginning of simulation, all nodes in the network are alive, so there is no difference between COTS and LEACH with respect to the number of packets accepted by BS. We can confirm it from the early slope of the two lines drawn as COTS and LEACH in Figure 10. At the middle phase of the simulation, the curve for COTS is more smoothly saturated. The decrease of the slope is based on the fixed frame size, and it shows that our approach works well. Later on in the simulation, the slope of COTS decreased then converged on zero. The longer time in COTS is thanks to the increased lifetime as shown in Figure 9.

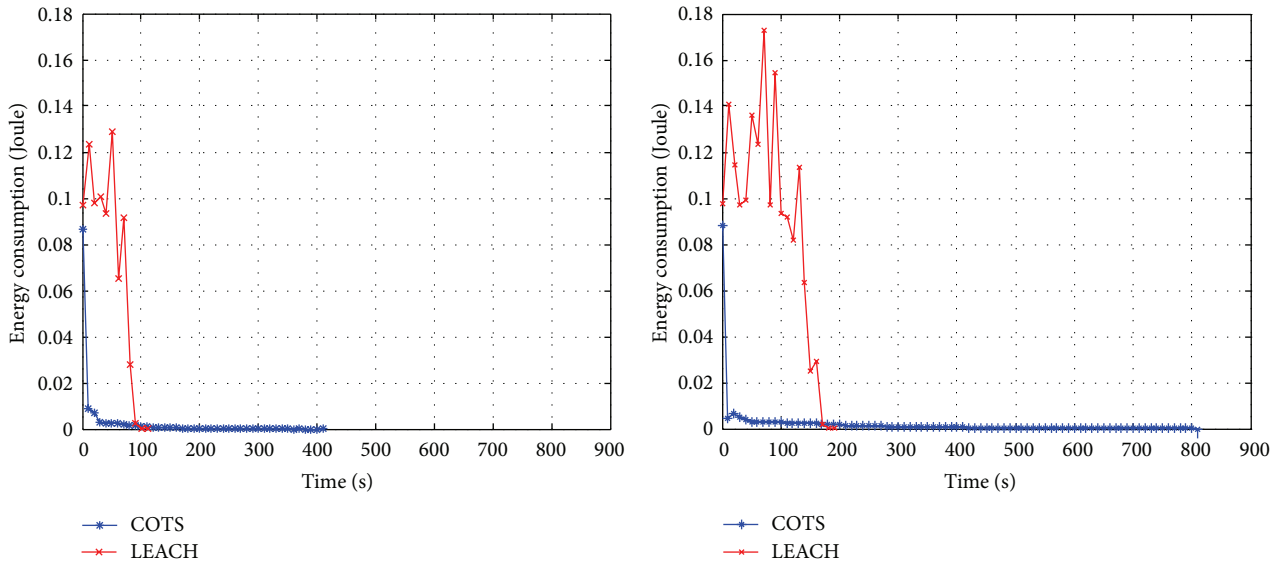
In Figure 10, we should pay attention to the relative height of the two graphs, but not the values. The height of the two figures is based on the accepted packets of LEACH. The vertical axis was divided into seven parts by the dotted lines. When the number of cluster heads (CHs) is small, the number of packets accepted by BS is slightly decreased in comparison to LEACH because the communication environment and parameters between sensor nodes (SNs) and their CHs are fixed at the first round and not averaged over the rounds resulting in some undelivered packets in COTS. Note that the cluster-reforming process may probabilistically average the communication environment and parameters between SNs and their CHs over the rounds in LEACH. However, when the number of CHs is large, the number of packets accepted by BS in COTS is more than that in LEACH as shown in Figure 13. Note here that the communication length is relatively short as the number of CHs increases, and, thus, the communication environment and parameters between SNs and their CHs will be more stable. Since the round interval is 10 seconds in these simulations, the system with 2 Joules has about two times as much as the simulation with 1 Joule, so the saved energy affects the total packets sent.

Figure 11 shows the total energy consumption per round. As shown in the figure, COTS consumes much less energy



(a) With an initial energy of 1 Joule (b) With an initial energy of 2 Joules

Figure 10: The number of packets accepted by BS (with 5 CHs).



(a) With an initial energy of 1 Joule (b) With an initial energy of 2 Joules

Figure 11: The total energy consumption per round.

than LEACH. As explained earlier, the setup phase is removed at every round after the first round, and just one time slot for rescheduling is added in COTS, resulting in significantly saved energy. Accordingly, except for the beginning of network lifetime, COTS consumes less energy than LEACH.

Figure 12 is the result of a simulation with 10 CHs that can be compared with Figure 9(b), which has 5 clusters. If the end of the network lifetime is defined as 90% of total nodes being dead, the end of lifetime of Figure 9(b) is 150 seconds, and the lifetime of Figure 12 is 300 seconds. The network in Figure 12 lives twice as long as that in Figure 9(b). This is because the fixed members in a cluster decrease flexibility as mentioned in Section 3.2. Increasing the number of clusters indicates the decreased area of each cluster. Therefore, the entire communication length also decreases, resulting in reduced bad effects. The reverse of the bad performance of Figure 9(b) is shown at 60 nodes, and the reverse of Figure 12 is at 80 nodes. Moreover, the staircase phenomenon comes more sharply into focus. It is caused by the energy-aware operation of CH at the steady-state phase. That is, CH stops when the energy of the battery is not enough to send a message to BS. The saved energy will be used at the rescheduling slot to guarantee an agreement of the CH orders among cluster

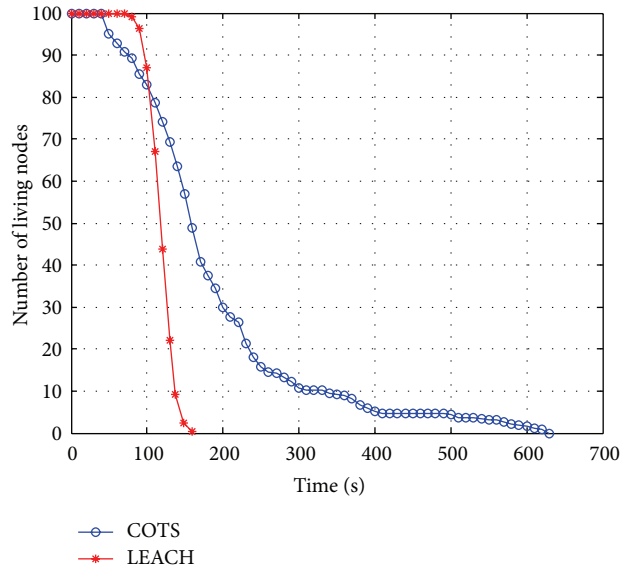


Figure 12: The number of living nodes (with 10 CHs).

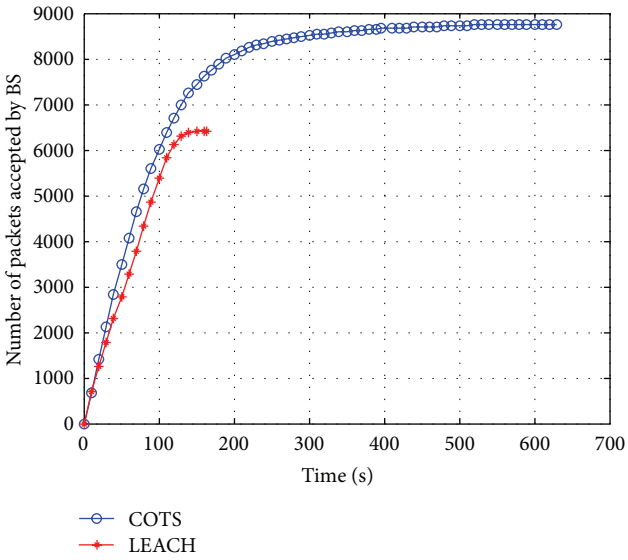


Figure 13: The number of packets accepted by BS (with 10 CHs).

members. It is a fact that the more the number of clusters is, the clearer this phenomenon will be.

Figure 13 draws the number of packets accepted by BS. We can realize that the higher CH allows COTS to show its advantages by comparing Figure 13 and Figure 10(b). The greatest strength is the relative increase of the total number of accepted packets. The performance of the accepted packets is reversed, when the network has ten CHs, and it is less than LEACHs when the network has five CHs. This suggests that the network saves more energy than before. It is somewhat surprising that the entire network lifetime is significantly improved even though the total number of packets is higher than that of LEACH during the network lifetime. As estimated from COTS's features, which are the static number of clusters and the data sent at regular intervals, the COTS has much higher reliability than LEACH. This is shown in the higher slope in the graph.

5. Conclusions

The proposed COTS significantly reduces energy consumption incurred by the setup phase of every round by realizing novel clustering without repeated setup, resulting in improved performance and prolonged network lifetime. Once a cluster is formed at the setup phase of the first round, the CH creates the cluster head order and broadcasts it to all the members in the cluster. As a result, the role of CH is rotated among members in a cluster without requiring any cluster-reforming process. The cluster head order is updated and announced to members at the rescheduling slot of every round, after it is checked against any dead member. The features of the COTS protocol can be summarized as follows.

(i) Some nodes die faster than others when using LEACH, but the entire lifetime is remarkably improved.

(ii) The more CHs there are, the less the lifetime is decreased.

(iii) The higher the initial energy is, the more energy is saved.

According to our simulation results, the network lifetime is prolonged more than 1.37 times in comparison to LEACH.

Acknowledgments

A preliminary version of this work was presented at the 4th International Conference on Computational Intelligence, Communication Systems and Networks, July 2012 [15]. The authors would like to thank the editor and anonymous reviewers for their helpful comments on this paper.

References

[1] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, vol. 3, no. 3, pp. 325–349, 2005.

[2] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: a survey," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6–27, 2004.

[3] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Annual Hawaii International Conference on System Siences (HICSS-33 '00)*, p. 223, January 2000.

[4] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.

[5] O. Younis and S. Fahmy, "Distributed clustering in ad-hoc sensor networks: a hybrid, energy-efficient approach," in *Proceedings of the 23rd AnnualJoint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, vol. 1, 2004.

[6] S. D. Muruganathan, D. C. F. Ma, R. I. Bhasin, and A. O. Fapojuwo, "A centralized energy-efficient routing protocol for wireless sensor networks," *IEEE Communications Magazine*, vol. 43, no. 3, pp. S8–S13, 2005.

[7] A. Manjeshwar and D. P. Agrawal, "TEEN: a routing protocol for enhanced efficiency in wireless sensor networks," in *Proceedings of the of 15th International Workshop on Parallel and Distributed Processing Symposium*, pp. 2009–2015, San Francisco, Calif, USA, 2001.

[8] A. Manjeshwar and D. P. Agarwal, "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 (IPDPS '02)*, pp. 195–202, 2002.

[9] N. Enami and R. A. Moghadam, "Energy based clustering self organizaing map protocol for extending wireless sensor networks lifetime and coverage," *Canadian Journal on Multimedia and Wireless Networks*, no. 4, pp. 42–54, 2010.

[10] K. Yadav, V. Pal, G. Singh, and R. Yadav, "Variable frame slot clustering scheme for data gathering wireless sensor networks," in *Proceedings of the Nirma University International Conference on Engineering (NUiCONE '11)*, pp. 1–5, Ahmedabad, Pakistan, December 2011.

[11] "UCB/LBNL/VINT Network Simulator—ns—2," 2012, http://www.isi.edu/nsnam/ns/.

[12] S. Oh, S. Hur, and G. Lee, "An efficient energy usage of wireless sensor network," *IE Interfaces*, vol. 23, no. 2, pp. 108–117, 2010.

[13] B. Shen, S. Zhang, and Y. Zhong, "Cluster-Based Routing Protocols for Wireless Sensor Networks," *Journal of Software*, vol. 17, no. 7, 2006.

[14] W. Bo, H. Han-Ying, and F. Wen, "An improved LEACH protocol for data gathering and aggregation in wireless sensor networks," in *Proceedings of the International Conference on Computer and Electrical Engineering, (ICCEE '08)*, pp. 398–401, Phuket, Thailand, December 2008.

[15] H. Shin, S. Moh, and I. Chung, "Energy-efficient clustering with one time setup for wireless sensor networks," in *Proceedings of the 4th International Conference on Computational intelligence, Communication Systems and Networks (CICSyN '12)*, pp. 64–69, Phuket, Thailand, 2012.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 345821, 11 pages
http://dx.doi.org/10.1155/2013/345821

Research Article

Adaptive Learning Based Scheduling in Multichannel Protocol for Energy-Efficient Data-Gathering Wireless Sensor Networks

Kieu-Ha Phung,[1,2] Bart Lemmens,[1,3] Mihail Mihaylov,[4] Lan Tran,[2] and Kris Steenhaut[1,3]

[1] *Department of Electronics and Informatics (ETRO), Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussel, Belgium*
[2] *School of Electronics and Telecommunications, Hanoi University of Science and Technology, Dai Co Viet 01, Hanoi, Vietnam*
[3] *Department IWT, Erasmushogeschool Brussel, Nijverheidskaai 170, 1070 Brussel, Belgium*
[4] *Faculty of Science and Bio-Engineering Sciences (DINF), Vrije Universiteit Brussel, Pleinlaan 02, 1050 Brussel, Belgium*

Correspondence should be addressed to Kieu-Ha Phung; kphung@etro.vub.ac.be

Received 26 October 2012; Accepted 8 January 2013

Academic Editor: Jiman Hong

Multichannel communication protocols have been developed to alleviate the effects of interference and consequently improve the network performance in wireless sensor networks requiring high bandwidth. In this paper, we propose a contention-free multichannel protocol to maximize network throughput while ensuring energy-efficient operation. Arguing that routing decisions influence to a large extent the network throughput, we formulate route selection and transmission scheduling as a joint problem and propose a Reinforcement Learning based scheduling algorithm to solve it in a distributed manner. The results of extensive simulation experiments show that the proposed solution not only provides a collision-free transmission schedule but also minimizes energy waste, which makes it appropriate for energy-constrained wireless sensor networks.

1. Introduction

Recently envisioned applications of wireless sensor networks (WSNs) exhibit more stringent Quality of Service demands in parallel with the traditional lifetime optimization problem. For example, Wireless Multimedia Sensor Networks (WMSNs), which enable new surveillance, traffic monitoring, and healthcare systems [1], clearly require large bandwidth. As the hardware costs are continuously decreasing, the number of WSN deployments is expected to grow drastically. Network density will increase greatly, resulting in high levels of inter-WSN interference. A new challenge needs to be addressed: the increased bandwidth demand in the presence of higher levels of interference.

The challenges stem from the fact that the bounded wireless channel capacity needs to be shared between nodes within the same geographical area. Meanwhile, most sensor nodes available on the market are equipped with IEEE 802.15.4 compliant transceivers which are able to operate on 16 nonoverlapping channels in the 2.4 GHz ISM band. Thus, using the available multiple channels efficiently to exploit parallel transmissions in WSNs becomes very attractive.

Although multichannel communication is quite well-investigated in wireless network [2–4], most proposals are not directly applicable to WSNs. Firstly, a sensor node is a low-power device with a single interface transceiver incapable to operate on several channels simultaneously, unlike other wireless devices such as laptop computers or personal digital assistants. As a consequence, a transmission will fail if the sender and its intended receiver are not on the same channel, which is called the deafness problem in multichannel communication. Secondly, dynamic channel negotiation or RTS/CTS (request to send/clear to send) exchange used in some approaches [5, 6] causes large overhead, as the size of WSN packets is much smaller (30 Bytes) than the packet size of the IEEE 802.11 standard (512 Bytes). Last, the bandwidth of sensor radios is limited compared to other wireless devices, for example, 50 Kbps to a maximum value of 250 Kbps.

Recently, several multichannel protocols have been designed specifically for WSNs. Fixed channel assignment approaches often divide the network into clusters or subtrees [7, 8] to which different channels are assigned to alleviate interference among clusters or subtrees. Those approaches cannot reduce interference efficiently because they do not

fully exploit the routing topology which has a large influence on the interference in the network. In [9], Yu et al. propose a game theoretic based channel assignment algorithm in which they take into account the routing information to reduce the interference more effectively but they do not fully achieve interference-free communication. Some other approaches schedule transmissions both in the time and frequency domain to achieve collision-free access [10, 11]. The scheduling algorithm proposed in [10] operates in a centralized manner, which causes extra overhead to communicate the assignment to each node. Moreover, they do not support heterogeneous traffic patterns.

A common application of WSNs is data gathering, where the many-to-one/converge-cast communication mode is usually taking place. In a multihop scenario, sensed data from nodes, called *source nodes*, are all relayed through one or several intermediate nodes, towards one or several base stations, called *sinks*. Therefore, the routing topology is often in the form of a tree or a forest, in which a node has one or several parent nodes. In fact, it is difficult to fully exploit routing information to achieve a collision-free channel assignment and transmission schedule since nodes must share information dynamically and be involved in the channel assignment or scheduling process. In such cases, the problem becomes NP-hard [9, 10].

Reinforcement Learning (RL) opens up a new path for practical WSN approaches. RL has already been applied to several problems in clustering, routing and neighbourhood management, medium access control, and radio duty cycling [12, 13]. It is a decentralized technique of goal-directed learning by trial and error [14]. Every node behaves as an autonomous agent, acting independently, obtaining the feedback from its environment, and learning as a result of this feedback. The feedback is in fact the result of the actions of all nodes in the system. Based on the obtained feedback, the agent adjusts its action in pursuit of its goal set by the system designer. In this paper, we aim at maximizing parallel transmissions to improve the network throughput by scheduling nodes' transmissions in order to avoid the collisions and the deafness problem in multichannel operation for data gathering WSNs. The elimination of the collisions and the deafness problem minimizes the energy consumption at each node. We show how sensor nodes learn to achieve successful transmissions and receptions in a decentralized manner, while minimizing energy consumption.

In this paper, we propose a multichannel protocol for data gathering WSNs with a Reinforcement Learning based scheduling algorithm. The proposed protocol exploits routing information for scheduling collision-free transmissions over multiple channels. The algorithm operates in a distributed manner and especially reduces the energy wasted by collision, idle listening, and deafness problem. Its performance has been evaluated through extensive simulations.

The rest of the paper is organized as follows: in Section 2, we present the background of the multichannel problem in energy-constrained WSNs and the formulation of the joint routing and scheduling problem in Section 3. The RL based scheduling algorithm and the details of the multichannel protocol are presented in Sections 4 and 5, respectively. In Section 6, the results of extensive experiments are presented to assess the performance of the proposed protocol. A discussion on the advantages and disadvantages of the proposed protocol is given in Section 7. Section 8 concludes this paper.

2. Preliminaries and Problem Formulation

2.1. Network Throughput and Multichannel Operation. In data gathering wireless sensor networks, the overall bound on the average network throughput per node is W/n, in which n is the number of source nodes and W is the transmission capacity [15]. Note that source and sink nodes are all equipped with half-duplex transceivers. However, it is proved that the maximum throughput can be reached only if the sink is 100% busy receiving packets and if the schedules of all nodes are aligned for interference-free communication for the given network topology. The main reason for the limitation on network throughput is *the interference in the shared wireless medium*. As depicted in Figure 1(a), while node 1 is transmitting to the sink, node 2 must be inactive and nodes 3, 4 can be inactive or communicate with each other on a different channel. Otherwise, they will create collision to the transmission between node 1 and the sink since they are all in the same collision domain.

Enabling interference-free spatial reuse of the shared wireless medium to improve network throughput has been an important research topic in the wireless networks domain [16]. There exist many different medium access control (MAC) methods designed for coordinating communication. The main techniques used to share the wireless medium and alleviate conflicts are contention based (carrier sense multiple access (CSMA)) or schedule based, such as time division multiple access (TDMA), frequency division multiple access (FDMA), and code division multiple access (CDMA). In addition, some approaches use power control and directional antennas to further reduce interference.

Figure 1(b) illustrates a schedule of TDMA collision-free transmissions for all source nodes to forward their data to the sink using a single channel/frequency for the line topology network in Figure 1(a). They are all in the same collision domain hence only one transmission can happen at one moment and the maximum achievable throughput is $W/5$. Higher network throughput can be achieved using multiple channels, depicted in Figure 1(c) (two channels in this case). Multichannel usage allows parallel transmissions within a single collision domain, which results in increased throughput, decreased collision probability, and thus better energy efficiency.

However, as the half-duplex transceivers of the sensor nodes are only capable of operating on a single channel at any time, a transmitter and a receiver must tune their radios to the same channel to communicate successfully. Given the same geographical network topology but different routing trees, the maximum achievable network throughput can differ as illustrated in Figure 2. When node 1 is chosen as the common parent of node 3 and node 4 for relaying their data, the maximum achievable network throughput is $W/5$, since node 1 cannot receive simultaneously from nodes

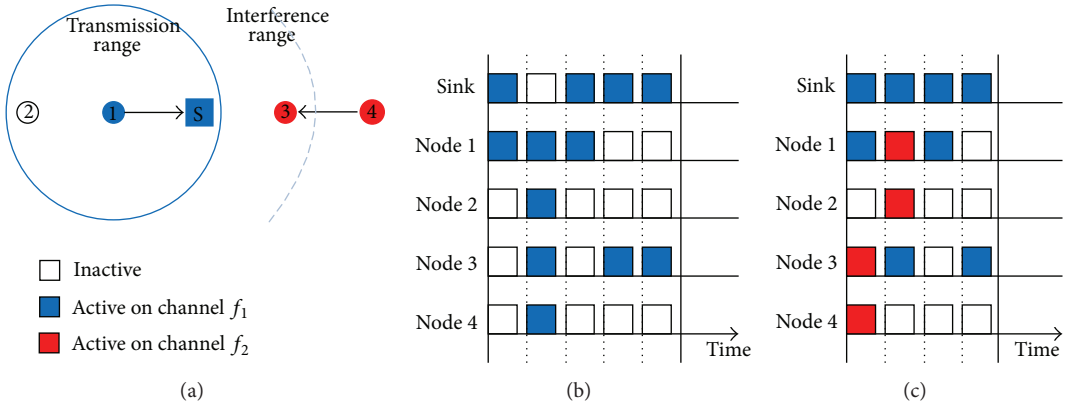


FIGURE 1: Network throughput, and collision-free transmission schedule in line topology. (a) Node 1 is transmitting to the sink, node 2 must be inactive on the same channel, and nodes 3, 4 are inactive or communicating on different channels. (The arrows point from sender to receiver.) (b) Time schedule of transmissions on single channel network. Each line depicts the operation of nodes. The achievable throughput is $W/5$ since it needs 5 time slots to transfer all data to the sink. (c) Time and frequency schedule of transmissions with two channels (the achievable throughput is $W/4$).

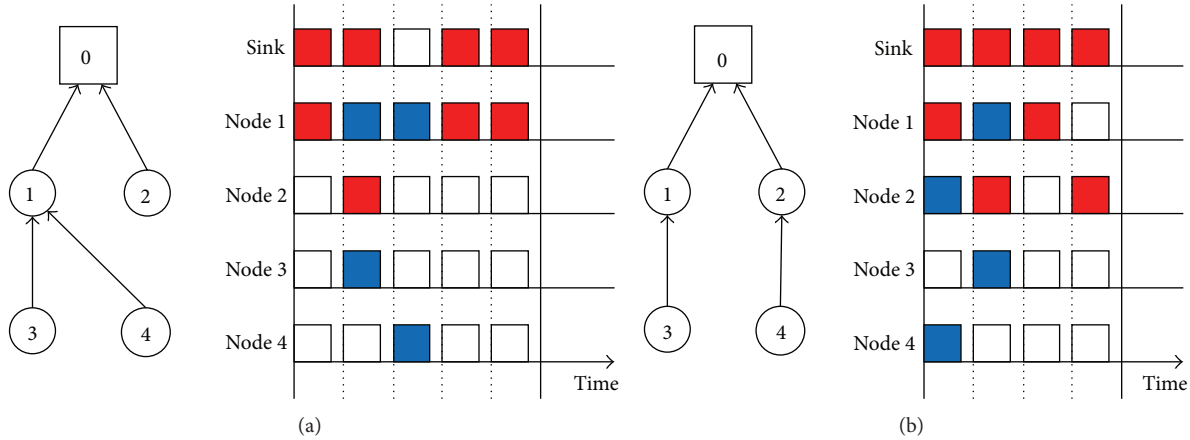


FIGURE 2: The influence of routing topology on achievable network throughput in the presence of half-duplex transceivers. In (a) a fixed routing structure is depicted in which nodes 3 and 4 have node 1 as relaying node. In this case, multichannel operation cannot increase throughput since node 1 can only receive or transmit on a single channel at any time. When nodes 3 and 4 select a different relaying node, as depicted in (b) the transmissions can be scheduled to maximize the number of concurrent parallel transmissions and the achievable throughput is $W/4$. (The arrows point from sender to receiver.)

3 and 4. In order to maximize the number of concurrent parallel transmissions, nodes 3 and 4 should forward their data to nodes 1 and 2, respectively. Therefore, it is possible to schedule node 1 transmitting to the sink while node 2 receiving packet from node 4 as depicted in Figure 2(b). It results in a maximum achievable network throughput of $W/4$. This proves that the routing topology can be managed for maximizing throughput.

2.2. Energy Efficiency and Channel Coordination. As can be seen in Figures 1 and 2, achieving a successful transmission between a pair of nodes in a multichannel wireless network requires the coordination between the sender and the receiver to transmit and to listen, respectively, at the same moment on the same channel. Furthermore, at that moment all other nodes in range must refrain from transmitting on that channel to avoid interference with the ongoing transmission. This coordination of transmissions provides opportunities to save energy at both the sender (no retransmissions due to *collision,* elimination of *hidden-terminal* and *deafness problem*) and at the receiver (no *idle listening* and *overhearing*).

The deafness problem occurs when a transmitter sends a control packet (e.g., request to send RTS) to initiate a transmission and the transceiver of the destination is tuned to another channel. After sending several requests, if the transmitter did not get any response (e.g., clear to send CTS) it may conclude that the receiver is not reachable anymore. Besides being confronted with the hidden-terminal problem as known in single channel, a multichannel operated network has to deal with the effect of the inability to receive control packets sent on a different channel than the one the node's

transceiver is currently tuned to. This phenomenon is called the multichannel hidden-terminal problem [3]. The nodes that miss the control packets might thus falsely conclude the channel is idle and start transmitting on that channel which will cause a collision. The overhearing of messages happens when a node receives messages that are destined to other nodes. The energy spent for receiving a message can even be a bit higher than the energy spent for transmitting a message [17]. These problems are the main sources of energy wasted in communication, which is the major contribution of energy consumption in WSNs, and shorten the lifetime of the network significantly.

The above-mentioned problems require the coordination of the transmissions between nodes. Several approaches have been proposed and [16] provides a detailed survey. For example, the dedicated control channel approach, a subset of dynamic channel assignment, uses a channel exclusively for control purposes [18]. On this control channel, nodes negotiate which channel to use for the actual data transmission. The dedicated control channel acts, however, as a bottleneck and results in a maximum achievable network throughput equal to the one of a single channel network. The frequency-hopping approach used in [5, 6] improves the network throughput by allowing parallel rendezvous transmissions without channel negotiation or information exchange. However, it encounters difficulties in dealing with the deafness and multichannel hidden-terminal problems. The well-known (Carrier Sense Multiple Access with Collision Avoidance) CSMA/CA handshaking process, RTS/CTS exchange, used in [5] cannot solve these problems completely. An RTS/CTS scheme is furthermore not suitable for WSNs due to the small packet size of IEEE 802.15.4, which would impose a large control overhead.

3. Model for Joint Routing and Transmission Scheduling

We aim to design a multichannel protocol that not only addresses the problems of collision, idle listening, overhearing, deafness, and hidden terminal, but also maximizes the network throughput. Taking into account the effect of the routing topology on the number of parallel transmissions, a node should select simultaneously to which parent to forward and on which channel to transmit. The problem becomes a *joint routing and transmission scheduling* problem.

The proposed protocol is based on a combination of TDMA and FDMA techniques. TDMA is known to provide collision-free operation and excellent energy efficiency due to the minimization of idle listening and overhearing. However, a practical implementation of TDMA in sensor networks is not trivial due to time synchronization requirements and scalability issues which will be discussed in Section 7.

Time is discretized into fixed length frames composed of a number of time slots. The length of a slot allows the transmission of a single data message and an acknowledgement message. Instead of using a fixed frequency assignment for each node, a channel-hopping scheme is used as follows.

(i) In each frame, a node periodically switches its channel at each time slot according to a chosen channel-hopping pattern, called the *default sequence*. A pseudo-random number generator is used to generate this sequence, in which the address of the node serves as the seed. In the sequence, there can be several broadcast slots in which all nodes hop to the same frequency. They can be used for local broadcast communication required by many upper-layer applications, for example, for updating a route metric.

(ii) In order to establish communication, the sender tunes its radio to the receiver's current channel and transmits the data. The sender can reproduce the listening channel of the receiver when it knows the receiver address. Therefore, it allows parallel transmissions between several pairs of nodes without exchanging information or negotiating a communication channel which pose a major challenge for energy and bandwidth constrained WSNs.

In each time slot, a node can be in three states, listening on its default channel, deviating to another channel to transmit data, or being radio-off to save energy resource.

We consider a multihop data-gathering network comprising a single base station and many sensor nodes sensing/monitoring data (e.g., environment parameters) which has to be forwarded to the base station over a routing tree. Each node has a nonempty set of parent nodes; otherwise, the node is disconnected from the network. The generated traffic at each node is assumed to be periodic with respect to the frame length, but the pattern may be different. Thus, heterogeneous traffic loads within the network are allowed. A collision occurs at a receiver when more than one node in the receiver's collision domain is transmitting on the channel the receiver is listening on. It also includes the hidden-terminal problem. For reliable communication, a transmission is considered successful if the sender receives the acknowledgement from the intended receiver.

Since a node normally has several parent nodes (each hopping according to their respective default frequency sequence), a node might have several channels to switch to. As illustrated in Figure 2, a fixed routing structure might prevent an optimal exploitation of the multiple channels. This problem can be addressed by letting the nodes decide which parent they should forward their data to depending on which channel the parent is listening. The selection of the parent hence may be different for each time slot.

During a frame, each node has a "pool" of actions it can perform. A node should specify the exact action, stay on its home channel, or deviate to a chosen channel to transmit in a certain time slot of the frame or withdraw from communication.

It is clear that without coordination, the transmission can fail because of collision, deafness problem, or (multichannel) hidden-terminal problem, see Section 2.2. The objective of the schedule algorithm is to coordinate the action of each node in each time slot to avoid as much as possible failed transmissions while forwarding all the generated data toward the sink.

4. Reinforcement Learning Based Scheduling Algorithm

The proposed scheduling algorithm is based on Reinforcement Learning (RL). Among machine learning methods, RL has shown to be suitable for WSNs since it is fully distributed and its implementation requires minimal memory space and communication overhead.

The sensor network is a multiagent learning system in which nodes can be considered as autonomous agents making distributed decisions, called *strategies*, on how to use the shared wireless medium. An agent learns from interactions with its environment, including the other agents, which strategies to play in order to improve its own long-term reward. When needed, a node has to compete with its neighbours for access to the shared radio medium. Moreover, the intended receiver node should be ready to listen on the same channel and thus not to be transmitting its own data. This translates to learning which action to perform, that is, *transmit*, *listen*, or refrain from communication (or *sleep*)—in each time slot. Through the *success* or *failure* of their actions, the nodes learn to adapt their strategies to coordinate the transmissions on the appropriate channels.

A formal description of our multiagent learning system consists of the following components.

(i) Each sensor node, denoted as n, is considered an autonomous agent that has a set of parent nodes, denoted as P_n, which can relay data of node n towards the sink.

(ii) The number of time slots in a frame is denoted as K and the current slot index is denoted as k, with $0 < k \leq K$.

(iii) A pseudorandom number generator function $c(n, k)$ is available to determine on which channel node n will listen in slot k.

(iv) In each slot k, node n can derive the set of default channels; its parent nodes will listen on $C_{nk} = \{c(p, k) \mid p \in P_n\}$.

(v) In each slot k, node n executes an action a_{nk} from its set of available actions A_{nk}: *listen* on its own channel $c(n, k)$ (denoted as action a_0 in Figure 3), or *transmit* on one of its parents' default channels c_x in C_{nk} (denoted as action a_x in Figure 3).

(vi) For each slot k, node n keeps track of the probabilities of successfully performing respective actions in that slot, $P_{\text{suc}}(a_x)$ for all $a_x \in A_{nk}$.

Note that the sink, equipped with a half-duplex transceiver as well, listens in each slot for incoming data.

The flowchart in Figure 3 depicts how the learning process operates in every slot. Initially, each node chooses an action at random for all slots. In each slot, the nodes perform their actions simultaneously and the combined actions of all nodes result in a specific state of the network. Some pairs of nodes might have successfully established communication, while the actions of other nodes are unsuccessful due to collisions or unresolved coordination. Note that the data

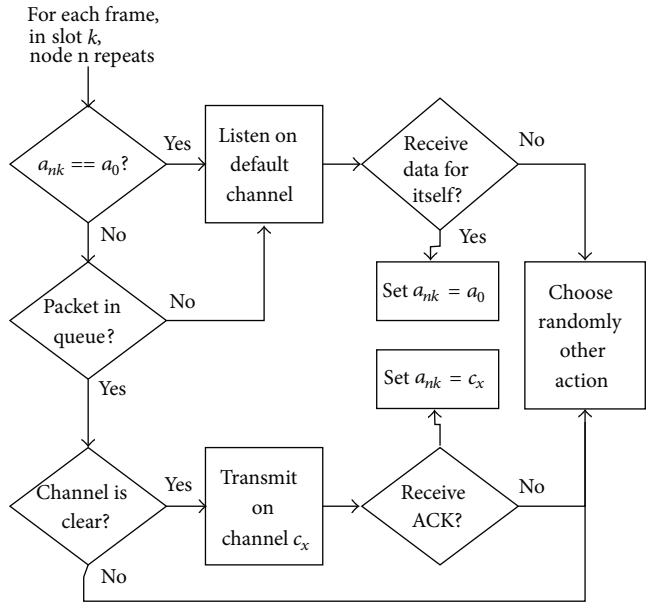


Figure 3: The learning process for scheduling collision-free transmission, which is operated in all nodes. The process is repeated every frame.

Table 1: Overview of why actions might fail.

Action	Reason for failure
Listen	Nobody sending (idle listening), collision
Transmit	Receiver not listening (deafness problem), collision (including hidden terminal)

mentioned in Figure 3 is the pseudo-data packet and the acknowledgement packet is the confirmation of receiving the pseudo-data packet.

When a node's action is successful, it will receive positive feedback either in the form of a data packet destined for itself (when listening) or an acknowledgment (when transmitting). The reason of a failed action is listed in Table 1.

To update the actions of the nodes, a trade-off between exploitation and exploration is made based on the "win-stay lose-shift" policy proposed in [19]. A successful action will be repeated in the same slot in the next frame ("win-stay"), while an action from the set of available actions will be randomly selected when the executed action failed ("lose-shift"). Different methods of action selection, based on a *uniform* or *biased* probability distribution, can be applied. In the uniform scheme, all actions in the set of available actions have equal selection probability. With the biased scheme, the probability of an action being chosen as the next action to perform is exponentially proportional to the probability of successfully performing that action (a parameter which has been updated in every frame). The policy used to balance exploitation and exploration influences to some extent on the optimality of the found solution.

To assure that nodes do not interfere with an already successfully coordinated communication, nodes perform a Clear Channel Assessment (CCA) check before transmitting.

At the beginning of each slot, a contention window is used in which the CCA is performed. The length of this contention window is decreased when a node has previously established successful transmissions in that slot. The mechanism will thus give privilege to nodes that performed successful transmissions before.

Individual nodes are not able to detect when the learning mechanism has converged, since it is fully distributed and nodes do not exchange their state information. The learning process is stopped after a fixed number of frames, set by the user of the system. After the learning process, the best action in each slot is chosen based on the success probabilities of the actions, in the following way:

$$\begin{aligned}
&\textit{select } a^* \textit{ with } P_{\text{suc}}(a^*) > P_{\text{suc}}(a_x) \quad \forall a_x \in A_{nk}.\\
&\textit{If } P_{\text{suc}}(a^*) > P_{\text{th}} \textit{ then}\\
&\qquad a_{nk} \longleftarrow a^*\\
&\textit{else}\\
&\qquad a_{nk} \longleftarrow \textit{sleep}\\
&\textit{end if}.
\end{aligned} \tag{1}$$

If the success probabilities of all actions are lower than or equal to a given threshold value (P_{th}), then the node infers that it cannot communicate successfully in that slot. The node can then turn off its radio in that slot, that is, *sleep*, to save its energy resource. The threshold value P_{th} is fixed by the user of the WSN.

5. Protocol Design

Time is discretized into fixed length frames composed of a specified number of slots. To achieve energy-efficient operation, the nodes duty cycle their radios in each slot. The length of the active period at the beginning of the slot is fixed and is long enough to allow a single IEEE 802.15.4 maximum length packet (of 128 bytes) to be transmitted and acknowledged. In every frame, each node periodically switches its channel at each time slot according to its pseudo-randomized channel-hopping sequence.

The protocol is divided into 3 phases as follows.

Initialization Phase. All nodes initially operate on the same default channel to allow initial synchronization and neighbour discovery. During network initialization, a minimum hop cost field is built. Each node learns its minimum hop distance to the sink and knows its set of candidate parent nodes through which it can relay data. If network density is high enough, this set will include more than one parent.

Scheduling Phase. Each node independently runs the learning process described in Section 4. Nodes learn how to coordinate their transmissions to achieve the goal of transferring the generated data message toward the sink while saving energy from failed transmissions and receptions. The achieved result is the schedule of actions (listen, transmit on a specified channel, or sleep) for all nodes in each time slot.

Operation Phase. According to the obtained schedule after the learning phase, nodes know whether they should turn the radio off or on and which channel to switch to. They all follow the strategies which the learning process converged to. The main advantage of the proposed scheme, the characteristic of Reinforcement Learning technique, is that nodes do not need to share their own schedule to others. The coordination among nodes is achieved without central authorization or completely sharing information.

The obtained nodes' schedule is considered an "optimal" solution when all the generated packets in a frame can be delivered successfully at the sink within a frame. The schedule of each node is traffic adaptive since nodes only contend for channel access when they have packets in their queue (see Figure 3). The traffic-adaptive mechanism can be illustrated by the behaviour of the leaf nodes. Since the leaf nodes do not have to forward any data, they only need to transmit in a specific number of slots according to their own traffic load. In all other slots, the leaf nodes will enter in *sleep* mode and turn their radios off in order to avoid idle listening and overhearing.

In the proposed protocol, nodes do not need to exchange RTS/CTS messages. Since a node either wins access to the channel (and hence does not need RTS/CTS) or fails and then attempts to find another free slot during the scheduling phase, nodes actively avoid the deafness and hidden-terminal problems.

6. Experimental Study

6.1. Evaluated Parameters. A Matlab simulation program was developed to evaluate the performance of the proposed multichannel protocol. The following metrics are measured.

(i) The end-to-end packet delivery ratio (PDR) is the ratio of the number of packets delivered at the sink to the total number of generated packets. This parameter indicates the optimality of the schedule obtained by the scheduling algorithm.

(ii) The end-to-end latency is the average of the end-to-end delay of all packets received by the sink. Combined with the end-to-end PDR, this indicates the achieved network throughput.

(iii) The energy waste factor is measured as the average amount of collisions, overhearing, and idle listening happening in one node per frame during the operation phase.

We compare the performance of our protocol with the multiple rendez-vous McMAC which is a frequency-hopping protocol.

We define the parameter α as the ratio of the number of time slots per frame to the number of generated data messages at all source nodes in the network. We investigate the relationship between the parameter α and the probability of converging to an optimal solution.

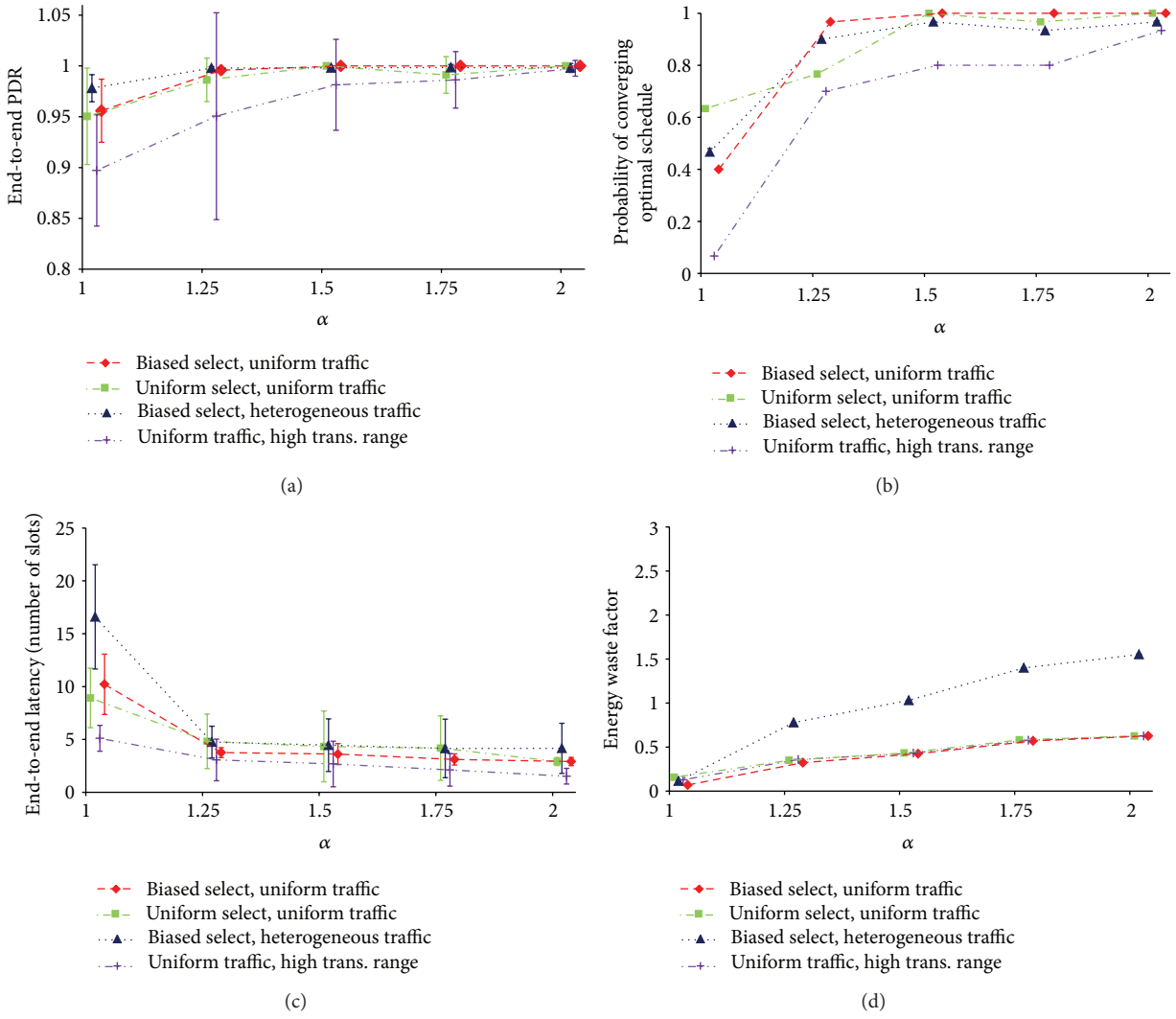


FIGURE 4: The performance evaluation in terms of (a) end-to-end packet delivery rate, (b) the probability of converging to optimal solution, (c) end-to-end latency, and (d) energy waste factor. The experiments have been done on grid topology of 25 nodes, in different settings, uniform or biased selection rule, uniform or heterogeneous traffic pattern, and different levels of transmission range.

We run simulation experiments with 25 nodes which are placed in an area of 80 m $*$ 80 m in two kinds of topologies, grid and random topology. The sink is positioned at the leftmost upper corner. Nodes are randomly deployed in the area or uniformly deployed at a distance of 20 meters from each other. We set the transmission range of nodes at two levels, 30 meters and 50 meters.

To illustrate the performance of the network at high data rates, we assume that each node generates messages every 10 seconds. The length of the frame is set to 10 seconds. Each node generates 1 message per frame (uniform pattern) or 1 to 4 messages per frame (heterogeneous pattern). Messages are generated at random time slots within frame.

The scheduling phase is set to 200 frames, which is long enough for the algorithm to converge. A good value for the success probability threshold P_{th} was empirically determined to be 0.4. The results are averaged over 30 simulation runs and are represented as function of the parameter α.

6.2. Evaluation. As illustrated in Figure 4(a), when α is increased, the average of end-to-end PDR is increased. In grid topology, uniform or heterogeneous traffic pattern, and biased selection rule, the average of end-to-end PDR is larger than 99.5% when α is larger than 1.25. Even though, the error bar of standard deviation is very small (less than 0.5%). The result indicates that in most simulation runs, a contention-free schedule capable of delivering all messages to the sink has been achieved by the RL based scheduling algorithm. The probability of converging to such optimal solution is nearly 100%, as depicted in Figure 4(b). When the number of time slots per frame is equal to the number of generated messages (α equals to 1), the nodes have less possibilities to arrange their transmission.

The high value of end-to-end PDR and high probability of converging to an optimal solution in heterogeneous traffic pattern show that the attained schedule is adaptive with the heterogeneous traffic demand of each node. It comes from

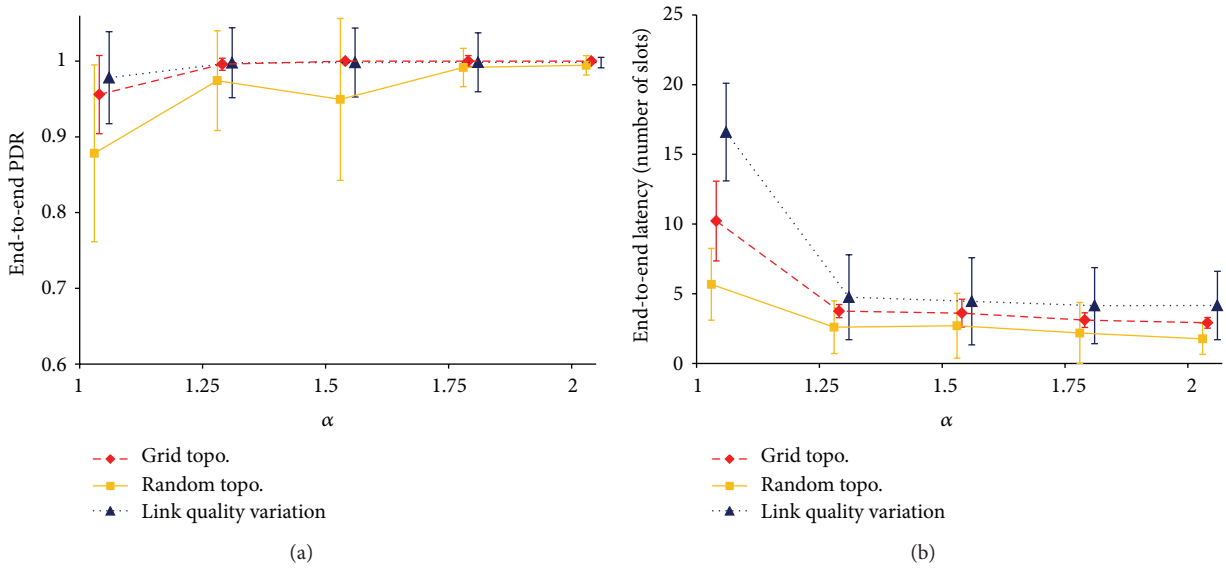


FIGURE 5: Performance evaluation of 25 nodes on random and grid topology and wireless link quality variation environment (average packet loss of 10%): (a) end-to-end packet delivery rate and (b) end-to-end latency.

the fact that a node only contends for the channel when it needs to send a message; otherwise, it listens or refrains from using the channel to save its own power and to give opportunity to others.

An increased communication range imposes a greater interference region as well. The nodes will thus have to overcome more constraints when they try to learn a collision-free schedule. Increasing the number of time slots per frame will alleviate this problem (when the number of parent nodes is about 3 to 5 in a grid topology with communication range of 50 meters, a good value for α is about 1.7).

The results of applying different rules for exploration, uniform or biased selection probability, show that the optimality of the obtained schedule is influenced by the exploration policy. With the uniform selection rule, the result is worse, lower average value and larger standard deviation of the end-to-end PDR parameter, which means that the attained solutions are often not optimal (not all nodes find the complete paths to send their messages). Figure 4(b) shows that the probability of converging to an optimal solution is a bit lower. The reason is that with a uniform selection rule, each action has equal probability to be chosen as an explored action. Nodes do not exploit the accumulated knowledge obtained during the learning process, which can be the number of successfully done trials of each action in the proposed scheme, which is used in biased probability selection rule. Therefore, the results are more fluctuating and not optimal.

Figure 4(c) shows the average number of time slots that a packet needs to be delivered to the sink. Equivalently to Figure 4(a), the performance is better when α is larger. For example, in grid topology, when the transmission range is set at 30 meters, the maximum hop count is 4 hops. The packets from the forth tier require 4 slots to be delivered to the sink while those from the first tier require only 1 slot, which is the optimal achievable latency.

The energy waste factor of each test scenario in Figure 4(d) illustrates the energy expenditure in useless activities of nodes. The higher the value of α is, the higher the energy waste becomes. In detail, nodes mainly waste their energy on the idle listening. The overhearing rarely happens and collisions are only found in nonoptimal schedules.

Figure 5 shows the performance comparison in different topologies, grid and random topology, and in link quality variation environment. In a random topology, a somewhat lower performance is perceived even when the number of slots per frame is high ($\alpha = 2$). The reason is that in random topology, the numbers of neighbour nodes and parent nodes are very different from node to node. Therefore, a fixed sleep threshold (P_{th}) appears unsuitable. We presume that a variable sleep threshold in relation with the number of parent nodes of each node will be more appropriate.

In real-world application, the quality of wireless link often varies, which means that the average of packet delivery rate at each link is less than 100%. We set up a simulation in which the average packet loss ratio of every link is 10%. To compensate for packet loss, the retransmission scheme is applied to assure reliable communication. In the learning phase, we set up each node to learn with the double of its own generated traffic. Hence, nodes contend more for access channel and might get more slots for transmissions. It would help nodes to have redundancy slots for retransmission if the transmission has failed because of low link quality. The effectiveness of the redundancy scheme for link variation has been proved by the very high end-to-end PDR and the one frame-bounded end-to-end delay illustrated in Figure 5.

The proposed protocol outperforms McMAC since McMAC does not provide contention-free access (see Figure 6). The receiver-centric contention resolution of McMAC is not enough to assure a decent packet delivery ratio. The average end-to-end latency is much higher because

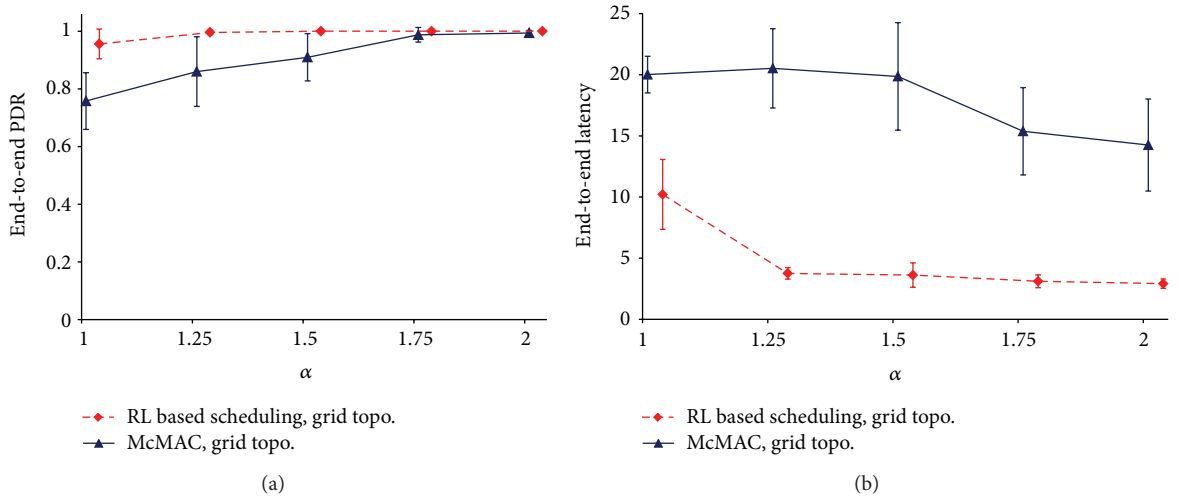


FIGURE 6: Comparison of Reinforcement Learning based scheduling in multichannel protocol with McMAC on grid topology of 25 nodes, traffic pattern of one packet per frame in terms of (a) End-to-end packet delivery rate and (b) End-to-end latency.

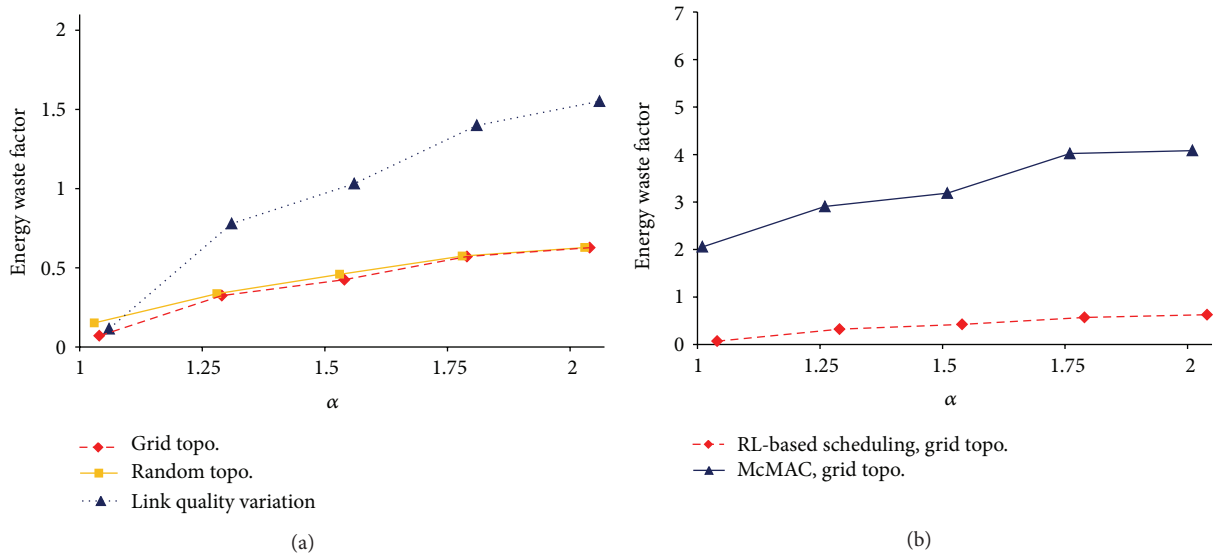


FIGURE 7: The energy waste factor of (a) Reinforcement Learning based scheduling in multichannel protocol on random or grid topology and link quality variation scenarios and (b) in comparison with McMAC.

of CSMA/CA scheme in McMAC. The proposed collision-free schedule protocol especially shows a very impressive energy efficiency compared to McMAC (see Figure 7(b)). For example, in McMAC protocol, on average a node has wasted its energy in 3 time slots every frame of 36 time slots (α = 1.5). In the proposed protocol, the result is 1 time slot every 3 frames, which is 9 times less than McMAC. In the adaptive learning scheduling protocol, most of energy waste comes from idle listening, while in McMAC, the collision, idle listening, and overhearing all happen much more frequently.

7. Discussion

The experimental results indicate that the performance of the proposed protocol depends on the parameter α, which is the ratio of the number of slots per frame to the number of generated data messages per frame. When α is large enough, the optimal solution is achieved more easily and more often, latency is lower and less packet loss is experienced. However, increasing α will result in more idle listening. When the number of slots per frame is larger, nodes might try to transmit in more slots before converging to a specified one to make some receiver node conclude that the action of *listening* is a good one. In addition, the conclusion of nodes also depends on the sleep threshold. To ensure a high packet delivery rate, the parameter of sleep threshold (P_{th}) should not be set too large; otherwise, a lot of nodes might decide to sleep. The value of P_{th} reported in Section 6 is obtained experimentally. Further investigation should be done to explore this issue.

The proposed protocol is based on schedule-based approach, which will require network-wide time synchronization which has been a well-investigated research topic in wireless sensor networks since it is essential for any kind of distributed components [20]. Among that, Time-Synch Protocol for Sensor Networks (TPSNs) [21] providing network-wide time synchronization with a master clock is emerging as a promising protocol.

In TPSN, the hierarchical-structured tree is used to distribute the global time of the master clock at the sink. Along the routing tree, nodes, receiving timing information, estimate their clock offset with the master clock and adjust their clock information when needed. Hence, nodes are time synchronized with the sink node. In [22], the experimental result on real testbed of a derivation of TPSN has proved the feasibility of tight requirement in synchronization for schedule-based approach protocol with reasonable overhead.

We observe that the operation of TPSN is suitable for our schedule-based multichannel protocol. The network is also built in a tree form in which each node knows its minimum distance to the sink. The timing information from the sink can be distributed to source nodes in broadcast slots. During periodic traffic forwarding to the sink, a sender node can receive timing information from the receiver which is its parent node in the routing tree and keep synchronized. Future work will present the detailed extension of TPSN for a practical time synchronization scheme supporting the proposed protocol operation on real hardware.

8. Conclusions

In this paper, we proposed a multichannel protocol for high bandwidth wireless sensor networks. The proposed protocol is based on the combination of TDMA and FDMA and provides collision-free scheduling of transmissions by applying a Reinforcement Learning technique for the joint scheduling and routing process in each node. The joint problem comes from the motivation that a fixed routing structure may result in suboptimal exploitation of the multiple channels. The medium access resolution is combined with route selection to make nodes learn not only to which parent but also on which channel they should forward their data. The scheduling algorithm works in a distributed manner on each node, resulting in an optimal solution of collision-free transmission for all nodes. Moreover, the obtained schedule is traffic adaptive and energy efficient.

We have investigated the performance of the proposed protocol with extensive simulation experiments. The results show that the optimal solution is obtained in nearly 100% of the simulation runs when the number of time slots per frame is greater than the traffic demand.

Taking into account wireless link quality variation, redundant traffic pattern learning is applied. It assures high end-to-end packet delivery rate and bounded end-to-end latency.

In comparison with McMAC, another frequency-hopping approach, our proposed protocol shows a better performance in terms of end-to-end delivery rate, end-to-end latency, and much higher energy efficiency.

In the future, we plan to implement the proposed protocol in combination with the appropriate time synchronization protocol on an operational sensor node testbed and evaluate its performance on the real devices.

Acknowledgments

This work was performed in the scope of FWO project G.0291.09N: QoS support in heterogeneous wireless sensor networks using distributed Reinforcement Learning techniques. K.-Ha. Phung is grateful to the Belgian Technical Cooperation (BTC) for granting her a mixed Ph.D. Scholarship.

References

[1] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "Wireless multimedia sensor networks: applications and testbeds," *Proceedings of the IEEE*, vol. 96, no. 10, pp. 1588–1605, 2008.

[2] A. Raniwala and T. C. Chiueh, "Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network," in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '05)*, pp. 2223–2234, Miami, Fla, USA, March 2005.

[3] J. So and N. Vaidya, "Multi-channel MAC for ad hoc networks: handling multi-channel hidden terminals using a single transceiver," in *Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MoBiHoc '04)*, pp. 222–233, Tokyo, Japan, May 2004.

[4] B. Vedantham, S. Kakumanu, S. Lakshmanan, and R. Sivakumar, "Component based channel assignment in single radio, multi-channel ad hoc networks," in *Proceedings of the12th Annual International Conference on Mobile Computing and Networking (MOBICOM '06)*, pp. 378–389, Los Angeles, Calif, USA, September 2006.

[5] P. Bahl, R. Chandra, and J. Dunagan, "SSCH: slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks," in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MobiCom '04)*, pp. 216–230, New York, NY, USA, October 2004.

[6] H. Sheung, W. So, J. Walrand, and J. M. McMAC, "A parallel rendez-vous multi-channel mac protocol," in *Proceedings of Wireless Communications and Networking Conference (WCNC '07)*, pp. 334–339, October 2007.

[7] G. Yu and C. Chang, "An efficient cluster-based multi-channel management protocol for wireless ad hoc networks," *Computer Communications*, vol. 30, no. 8, pp. 1742–1753, 2007.

[8] Y. Wu, J. Stankovic, T. He, and S. Lin, "Realistic and efficient multi-channel communications in wireless sensor networks," in *Proceedings of the 27th IEEE Conference on Computer Communications (INFOCOM '08)*, pp. 1193–1201, 2008.

[9] Q. Yu, J. Chen, Y. Fan, X. Shen, and Y. Sun, "Multi-channel assignment in wireless sensor networks: a game theoretic approach," in *Procedings of the 29th IEEE Conference on Computer Communications (INFOCOM '10)*, pp. 1–9, March 2010.

[10] M. Salajegheh, H. Soroush, and A. Kalis, "HYMAC: Hybrid TDMA/FDMA medium access control protocol for wireless sensor networks," in *Proceedings of the 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '07)*, pp. 1–5, September 2007.

[11] O. D. Incel, L. van Hoesel, P. Jansen, and P. Havinga, "MC-LMAC: a multi-channel MAC protocol for wireless sensor networks," *Ad Hoc Networks*, vol. 9, no. 1, pp. 73–94, 2011.

[12] A. Förster and A. L. Murphy, "Froms: a failure tolerant and mobility enabled multicast routing paradigm with reinforcement learning for WSNs," *Ad Hoc Networks*, vol. 9, no. 5, pp. 940–965, 2011.

[13] M. Mihaylov, Y. A. le Borgne, K. Tuyls, and A. Nowé, "Decentralised reinforcement learning for energy-efficient scheduling in wireless sensor networks," *International Journal of Communication Networks and Distributed Systems*, vol. 9, pp. 207–224, 2012.

[14] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*, The MIT Press, Cambridge, Mass, USA, 1998.

[15] E. J. Duarte-Melo and M. Liu, "Data-gathering wireless sensor networks: organization and capacity," *Computer Networks*, vol. 43, no. 4, pp. 519–537, 2003.

[16] O. D. Incel, "A survey on multi-channel communication in wireless sensor networks," *Computer Networks*, vol. 55, no. 13, pp. 3081–3099, 2011.

[17] http://www.ti.com/product/cc2420.

[18] J. Li, Z. Haas, M. Sheng, and Y. Chen, "Performance evaluation of modified IEEE 802.11 MAC for multi-channel multi-hop ad hoc network," in *Proceedings of the 17th International Conference on Advanced Information Networking and Applications (AINA '03)*, pp. 312–317, 2003.

[19] M. Posch, "Win-stay, lose-shift strategies for repeated games—memory length, aspiration levels and noise," *Journal of Theoretical Biology*, vol. 198, no. 2, pp. 183–195, 1999.

[20] I. K. Rhee, J. Lee, J. Kim, E. Serpedin, and Y. C. Wu, "Clock synchronization in wireless sensor networks: an overview," *Sensors*, vol. 9, no. 1, pp. 56–85, 2009.

[21] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, pp. 138–149, Los Angeles, Calif, USA, November 2003.

[22] B. Lemmens, K. Steenhaut, P. Ruckebusch, I. Moerman, and A. Nowe, "Network-wide synchronization in wireless sensor network," in *Proceedings of 19th IEEE Symposium on Communication and Vehicle Technology in the Benelux (SCVT '12)*, Eindhoven, The Netherlands, November 2012.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 421084, 13 pages
http://dx.doi.org/10.1155/2013/421084

Review Article

Intelligent Optimization of Wireless Sensor Networks through Bio-Inspired Computing: Survey and Future Directions

Sohail Jabbar, Rabia Iram, Abid Ali Minhas, Imran Shafi, Shehzad Khalid, and Muqeet Ahmad

Bahria University Wireless Research Center, Bahria University, 44220 Islamabad, Pakistan

Correspondence should be addressed to Sohail Jabbar; sjabbar.research@gmail.com

Received 14 July 2012; Revised 15 December 2012; Accepted 17 December 2012

Academic Editor: Jiman Hong

This survey article is a comprehensive discussion on Intelligent Optimization of Wireless Sensor Networks through Bio-Inspired Computing. The marvelous perfection of biological systems and its different aspects for optimized solutions for non-biological problems is presented here in detail. In the current research inclination, hiring of biological solutions to solve and optimize different aspects of artificial systems' problems has been shaped into an important field with the name of bio-inspired computing. We have tabulated the exploitation of key constituents of biological system for developing bio-inspired systems to represent its importance and emergence in problem solving trends. We have presented how the metaphoric relationship is developed between the two biological and non-biological systems by quoting an example of relationship between prevailing wireless system and the natural system. Interdisciplinary research is playing a splendid contribution for various problems' solving. The process of combining the individuals' output to form a single problem solving solution is depicted in three-stage ensemble design. Also the hybrid solutions from computational intelligence-based optimization are elongated to demonstrate the emergent involvement of these inspired systems with rich references for the interested readers. It is concluded that these perfect creations have remedies for most of the problems in non-biological system.

1. Introduction

After exhaustive effort in gaining maximum productivity from brainstorming solutions, researchers are now inclining towards biologically inspired solutions. Each and every creation of the One has the invitation to the researchers and thinkers to ponder. This creation is perfect in its own life and needs. The ultimate ideological convergence of scientists to this perfect creation for finding the relevant traces of problems and their solutions is in fact moving to "the best." Giving insight attention to the nature and the creation gives the glow to thoughts due to its miraculous architecture. Ibn al-Haitham, a 10th-century mathematician, astronomer, and physicist, invented the first pin-hole camera by understanding the mechanism that light enters the eye, rather than leaving it [1]. In the 9th century, Abbas ibn Firnas was the first person to make a real attempt to construct a flying machine and fly. He designed a winged apparatus, roughly resembling a bird costume [2]. Moreover, Newton's theory of gravity, Wright brothers' influenced work of flying, Einstein's theory of relativity, and John von Neumann's abstract model of self-reproduction are just the names of few. The intense exploitation of these miraculous creations of the One in Human's problems solving has now organized as a complete field naming biological-inspired computing. The word computing encompasses the activities of using and improving the computer technology, computer hardware, and computer software. The technique of formulating the views, ideas, technologies, and algorithms for unknotting the issues as well as introducing the innovation in computer technology, hardware, and software that are motivated by the biological systems is called biologically inspired computing or bio-inspired computing. Its typical consortium is shown in Tables 1 and 2. Three key subfields are loosely knit together to form the core of bio-inspired computing which are connectionism, social behavior, and emergence. Though biological systems exhibit complex, intelligent, and organize behavior, they are comprised of simple elements governed by simple rules. It is highly cost effective to hire the complex and in-networked processing-based simple rules in man-made systems. The real

effort that comes in this synergistic solution is in finding the similar patterns of issues in artificial (man-made) and real (natural) creation and later their solutions. This invites the interdisciplinary research which is an essential driver for innovation.

The rest of the paper is organized as follows. Section 2 summarizes the intelligent optimization in biological systems. Bio-inspired intelligent optimization for non-biological system is discussed in Section 3. Section 4 shows the optimization in wireless sensor networks' domain. In continuation to this, in Section 5 a connection is tried to be established between real and artificial systems to show how different artificial systems have been developed from the real system and what are the constituents of this establishment. Solution of various problems from the wireless sensor networks' domain from different bio-inspired systems is discussed and tabulated in Section 6. Future direction and conclusion are given in the subsequent sections.

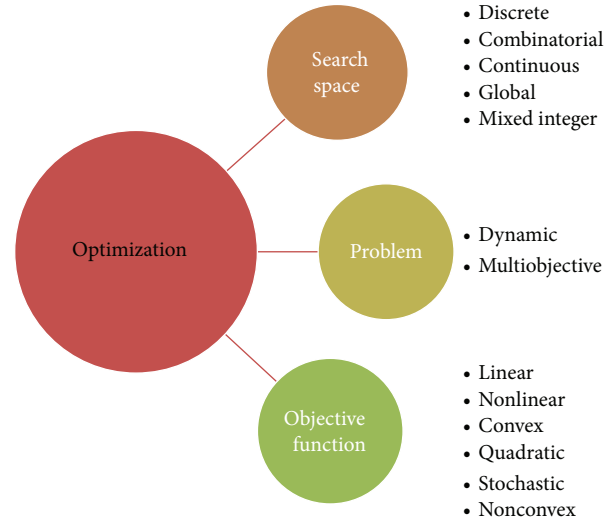


Figure 1: Optimization categories.

2. Intelligent Optimization in Biological Systems

The real creation of any natural system has marvelous traits in it. The source of inspiration is the traits which provide a guiding light to lead us towards the optimal solution for inspired systems. As the traditional optimization problems need more memory and computational power and are directly exponentially related with the problem size, optimization can be an interdisciplinary approach for finding the best solution of a problem among alternatives. Better computer systems and improved performance of computational tasks such as flexibility, adaptability, decentralization, and fault tolerance can be achieved in the form of bio-inspired algorithms from the principles evolved from Nature. As far as modeling of a problem with respect to the bio-inspired algorithms is concerned it enables the problem to solve according to the principles evolved from nature by understanding nature's rule the complex problems. Figure 1 below shows the optimization techniques classification to solve optimization issues based on three categories, that is, search space, problem, and form of objective function.

One of the key features of biological systems in searching the best solution is optimization. Most of the time, optimization process needs iteration of working sessions. Foraging behavior in ants and bees systems, flocking, herding, and schooling process in birds, animals, and fish, respectively, are some of the typical examples of iteration of working sessions for the optimal respective solutions. Optimization is the scientific discipline that raises the winning flag for the solution which gives the optimal result among all available alternatives in resolving the particular problem. Swarm intelligence (SI) is one of the aspiring solutions from bio-inspired computing for such heuristic optimization problems, for example, routing. Swarm refers to the large no. of insects or other small organized entities, especially when they are in motion. Global intelligent behavior to which the individuals are entirely unknown is emerged due to the local, self-organized, and decentralized interaction of swarm's agents. Natural examples of SI include ant colony, bird flocking, animal herding, bacterial growth, fish schooling, drop water, and fireflies. Examples of algorithms under the head of SI are Ant Colony Optimization (ACO), River Formation Dynamics (RFD), Particle Swarm Optimization (PSO), Gravitational Search Algorithm (GSA), Intelligent Water Drop (IWD), Charged System Search (CSS) and Stochastic Diffusion Search (SDS). Among all genus under swarm intelligence, autonomously social and extremely well-organized as well as colony-based life is of Ants. Inspired algorithm from ants is Ant Colony Optimization. Number of combinatorial optimization problems are targeted by the ACO technique such as asymmetric travelling salesman [3], graph coloring problem [4], and load balancing in telecommunication networks [5]. Ant Colony Optimization has been formalized into a meta-heuristic for combinatorial optimization problem. A meta-heuristic is a set of algorithmic concepts that can be used to define heuristic methods applicable to wide set of different problems. Examples of meta-heuristic include tabu search [6–8], simulated annealing [9, 10], iterative local search [11], evolutionary computation [12–15], and ant colony optimization [16–19]. For their applications in different fields for the optimal solutions, their nature is customized accordingly. For example, in case of Wireless Sensor Network (WSN), which is an energy, memory, and computational power constraint network, the proposed heuristic algorithm must be customized enough to cater for these constraints.

3. Bio-Inspired Intelligent Optimization for Nonbiological System

Survival of the best is the key driving force behind the optimization in artificial systems. From nothing to existence of global village clearly portrays this endeavor of moving to "the best." The influence of biology and life sciences is a source of inspiration since ages, as the design of aircraft leads to be inspired from birds, the fighter jet planes from

Table 1: Optimization in wireless sensor networks' domain.

Optimization problem	WSN challenges	Objectives of optimization problems in WSN
Multiobjective [20]	*Design and deployment*	*Multitude objectives of deployment/design issues of WSN like coverage, differential detection levels, lifetime of node, energy-related parameters, energy efficiency, sensing point uniformity, and number of sensors which needs to be optimized simultaneously or optimal tradeoff is required between objectives.*
Multidimension [24]	Localization	Deals with position estimation, signal propagation time, and the received signal strength of which at least one nonlinear function needs to be optimized.
Combinatorial [23]	*Security*	*Searching for optimal solution from discrete variables of the functions like node capabilities, possible inside threats, integrity, end-to-end data security, and key distribution for the best solution.*
Multi objective [21]	Routing and clustering	Multitude objectives of routing and clustering of WSN like energy consumption, network delay, and data packet lost rate, helps relay the sensed data traffic to the sink, which needs to be optimized simultaneously or optimal tradeoff is required between objectives.
Combinatorial [25]	*Data aggregation*	*The optimal solutions may be discrete or it may be discrete solution like efficient collection of sensed data, forwarding data to base station, and integrating data derived from multiple sources for the best solution.*
Multi objective [22]	Scheduling and MAC	Multitude objectives passive mode, active mode, and transmission of data by node for maximum life time by conserving energy, which needs to be optimized simultaneously or optimal tradeoff is required between objectives.
Combinatorial [26]	*QoS management*	*The optimal solutions may be discrete or it may be discrete solution like analytically plan and allocate resources for the best solution.*

Table 2: Exploitation of key constituents of biological systems for developing bio-inspired systems.

Key constituents of biological system	Examples of constituents	Bio-inspired system
Cell	*Structural unit*	*- DNA computing [27]* *- Membrane computing [28]* *- Cellular automata [29]* *- Rendering [30]*
Organ	- Arteries - Bronchi - Ears - Heart	- Sensor network [31] - Excitable media [32]
Organ system	*- Circulatory system* *- Digestive system* *- Nervous system* *- Reproductive system* *- Respiratory system* *- Skeletal system* *- Urinary system*	*- Artificial neural system [33]* *- Artificial immune system [34]* *- Genetic algorithm [35]* *- Biological fault tolerant [36]*
Organism	- Animals - Plants - Protists - Bacteria	- Fractal geometry [37] - Lindenmayer system [38] - Morphogenesis system [39] - Artificial life [40]
Ecosystem	*Biological system and its physical environment*	*- Biodegradability production [41]* *- Emergent systems [42]* *- Communication network and protocol [43]*

birds of prey, commercial jet planes from the birds that travel long distances, submarines and other deep sea diving machines from deep sea fish, surgical tools from beaks of various birds, helicopters from dragon flies, Bionic arms, and muscles from human arm and muscles. The effort in terms of time and space of finding the optimal solution depends on application, search space and solution required optimality level. The alternative prospective feasible solutions of that specific problem are reduced if the constraints are posed by the user or the problem itself.

The constraints usually restrict the search space. The presented problems where the constraints are absent are called it unconstrained optimization. In contrast to constraint optimization problem where the limits and boundaries are defined, global optimization problems is concerned with the detection of optimal one. This is not always possible or

necessary indeed. There are cases where suboptimal solutions are acceptable depending on their quality compared to the optimal one. This is usually described on local optimization. Conventional approaches exist to achieve the better solutions but the research is now inclining towards the bio-inspired intelligent optimization techniques. Artificial Neural Network (ANN) is a massively parallel computing systems consisting of large number of interconnected simple processors to handle various types of challenging computational problem such as in wireless sensor network, RF and microwave communication, Radar communication, and Antenna and wave propagation. This aspiring system is based on the organizational principles used in humans. Rendering/computer graphics technology is inspired by the pattern of animal skins, bird feathers, bacterial colonies, and mollusk shells which have their applications in pattern recognition, wireless sensor network, and others. Similarly, inspiration of artificial immune system by natural immune system, fractal geometry by clouds, river networks, and snowflakes are some examples of it. The next section clearly intuits the emergence of bio-inspired solution for the optimization in various applications of wireless sensor network.

4. Optimization in Wireless Sensor Networks' Domain

In most of the real-world problems, an optimal solution for simultaneously satisfying the one or more conflicting performance attributes is required. The key in seeking the optimization in solution for such type of problems is setting tradeoff among the participating attributes in their solutions. For example, in aircraft design, tradeoff is set between minimizing the fuel consumption and maximizing its performance, and real-time communication of sensed data from source to destination, the congestion, packet drop and the energy consumption are also minimized as the participating attributes. In [20–22], Authors have targeted the wireless sensor networks' domain for the optimization of multitude objectives in design and deployment, routing and clustering, and scheduling and MAC aspects. In [20] the author has analyzed the major challenges in designing wireless sensor network. He has presented the multiobjective deployment of nodes with establishing the tradeoff between the desired and contrary requirements for coverage, differentiated detection levels, network connectivity, network life time, data fidelity, energy efficiency, number of nodes, fault tolerance, and load balancing.

He has also purposefully discussed the need for simultaneous optimization of the aforementioned objectives. Wei and Zhi [21] have introduced an advanced ant colony algorithm based on cloud model (AACOCM) for the optimization of multitude aspects of achieving optimal routing. This multiobjective optimization model optimizes the functions such as energy consumption, network delay, and data packet loss rate. Various services having different energy, cost, delay, and packet loss requirements can be focused by tuning the weight of each function of this proposed model. Passive mode, active mode, and transmission of data by node for maximum life time by conserving energy are the multitude objectives that are focused in [22], which needs to be optimized simultaneously or optimal tradeoff is required between these objectives for the purpose of achieving better scheduling and accessing the media at MAC layer.

In Combinatorial Optimization (CO) some or all of the participating objects or variables are discrete or restricted to be integral. The target in CO is to find an optimal object where finite objects are available. Vehicle Routing Problem (VRP), Travelling Salesmen Problem (TSP), Minimum Spanning Tree (MST) problem, and data routing in WSN are its typical examples. Tie and Li [23] have introduced a security key management scheme which is based on combinatorial optimization. The results come up with yielding optimal results for the number re-key messages. The ultimate target is to search for the optimal solution for discrete variables of the function like node capabilities, possible inside threats, integrity, end-to-end data security, and key distribution for the best solution.

5. From Real to Artificial

Some of the things are similar to each other at some important aspect even of their opposed nature. A desktop computer is a scooped-out hole in the beach where information from the cyber sphere wells up like sea water is a typical example of developing similarity between non alike things. But the key in making such relationship needs to explore the nature of two comparing systems. In the following paragraphs, Ant as a typical example of biological system is explored with respects to the basic constituents of biological systems. The foraging behavior of Ants is taken to develop the relationship with the routing in Wireless Sensor Networks. A typical representation of ACO algorithm with respect to Ants foraging behavior is shown in Figure 2.

The constituents of biological systems which forms the basis for developing metaphoric relationship to artificial system are cell (structural unit), organs (cell joins to form tissues and tissues joins to form a functional body), organic system (functional body interacts to work for same task), organism (multiple organic systems join to form a complete structure), and ecosystem (environment interacting to the complete structure). The perfection of the real system from unit to system is reflected in Table 2 by showing the emergence of these components for the exploitation in developing inspired systems. Many complete research fields inspired by the constituents of biological system have been established, which have their key roles in optimizing the solution for various problems in non-biological systems. The indispensable role of bio-inspired solutions in variety of applications for their optimized solution is demonstrated in the subsequent paragraphs.

Architecture in interlinking of individuals matters a lot in building the power of unity. In case of centralized systems reliable, effective, and efficient coordination to the central hub is indispensable. Death of central point is the end of the system. In contrast to it, decentralized architecture needs powerful autonomous entities on one hand, and

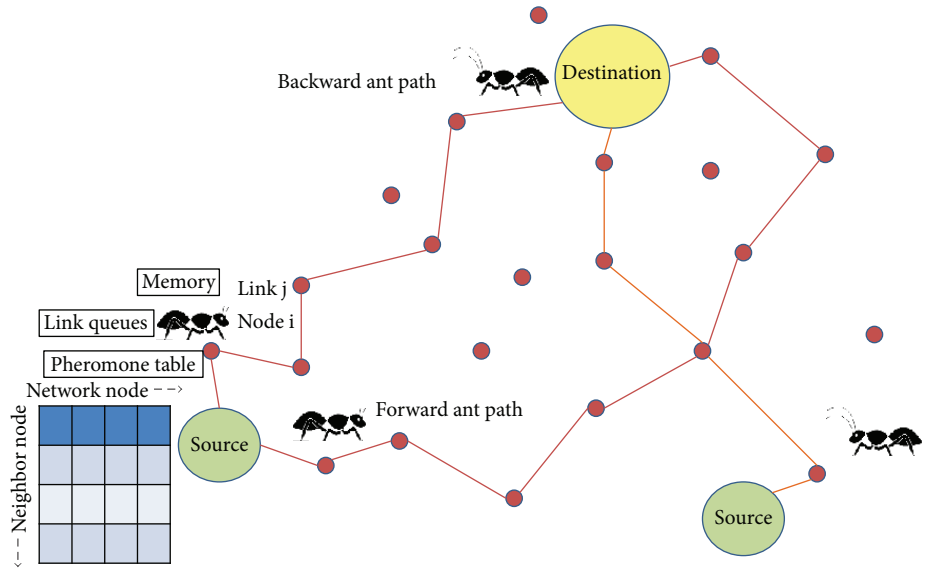


Figure 2: Representation of ACO Algorithm With Respect to Ants Foraging Behavior.

Table 3: Application of bio-inspired systems in wireless sensor network.

Inspired systems	Natural systems	Wireless sensor network application(s)
Artificial neural network [33]	*The brain*	*Power efficiency [44], fault detection [45], self-configuration [46]*
Artificial immune network [34]	Immune system [47]	Misbehavior detection [48], image pattern recognition [49]
Genetic algorithm [35]	*Natural evolution system*	*Dynamic shortest path routing [50], lifetime maximization [51]*
Cellular automata [29]	Life Life like/ game of life	Large network simulations [52], area coverage scheme [53]
Rendering (computer graphics) [30]	*Patterns of animal skins, birds feathers, mollusk shells, and bacterial colonies [54]*	*Range-free localization [55]*
Fractal geometry	Clouds, river networks, snowflakes, cauliflower or broccoli, and systems of blood vessels and pulmonary vessels, ocean waves	Antenna designing [56]
Communication networks and protocols	*Epidemiology*	*Cross-layer communication protocol [57, 58], underwater communication [59]*
Excitable media	Forest fires, wave, heart condition	Designing of intelligent sensor systems [60]
Sensor network	*Sensory organs*	*Self-configuration [61], target tracking [62]*

then it also reflects the behavior of adaptation (flexible to changing environment), self-assembly (unit to be one), self-organizing (interaction for the one), and self-regulation, (keeping the process tunably smooth) [25] as in WSN due to the distributed knowledge, distributed control and scalable properties on the other hand. Although the architecture is application dependent, decentralized system has variety of advantages over the centralized system. The real creation of the One also exhibits such fascinating behavioral and architectural traits to which Brownlee has pointed out in [71]. He has quoted architectural and behavioral characteristics in biological system with referring to Lodding and Nas [72] and Marrow [73]. Typical behavioral traits are adaptation, autonomous, fast and appropriate solutions in the short term, specific and optimized solutions in the long term, memory, self-assembly, self-regulation, and self-organization, learning, and intelligence. This is in parallel to architectural traits in natural systems such as decentralized control, distributed knowledge, emergent behavior dynamic, fault and noise tolerant, flexibility, parallel processing, persistent, robustness, and scalability. Insight study in finding the similar traits is the prerequisite to develop the metaphoric relationship between the two natural and artificial systems. Iram et al. [74] have discussed the above-mentioned traits in the context

of developing the metaphoric relationship between the two systems.

6. Wireless Sensor Networks and Bio-Inspired Systems

In this section, we have taken Ant Colony Optimization inspired by Ant System (AS) from the biological field and WSN from the artificial systems to understand the metaphoric relationship between the natural and artificial systems. Firstly, we have bridged the gap between the two technologies by finding similar traits. Then foraging behavior of ants and the routing behavior in WSN are analyzed to develop their metaphoric relationship.

In AS there is no central dictation for the target and even of the next goal. Rather stigmergic communication through pheromone is the only source dictating the ants for any action, that is, foraging, invading, security, mating, migration. Ants are distributed on the field and follow each other through the traces of pheromone trails. If ants are considered vertices then the pheromone trails between them represent the edges forming a graph theoretical problem. Due to the decentralized and distributed nature of ants in foraging between the source and destination, the algorithmic complexity of their foraging behavior (routing) turns to the worst case of $O(2^n)$, that is, exponential making it an NP hard problem as their first time exact and optimal solution is almost impossible. This meta-heuristic nature of foraging behavior's solution needs recursion of searching process for optimal result. Complete participation of Ant's swarm with multi-objectives of rich food source, shortest and safe path, making the foraging process collectively as multi-objective combinatorial optimization problem.

Among the wireless networks, wireless sensor network is the most emerging and multiconstraint (energy, memory, and computational as well as bandwidth) technology. There also come up some differences from the traditional wireless networks like cellular network and especially from its close ancestor, ANet (Ad hoc Network) like in-attendant operation for a long period in the energy constraint environment, densely and randomly deployed nodes, high node failure, and other constraint resources. WSN has its applications in wildlife monitoring, cold chain monitoring, Glacier monitoring, rescue of avalanche victims, cattle herding, geographical monitoring, monitoring of structures, vital sign monitoring, ocean water and bed monitoring, monitoring of fresh water quality, tracking vehicles, sniper localization, volcano monitoring and tunnel monitoring, and rescue. Underwater sensor network, that is typically based on ultrasound, is also a key application of WSN [75, 76]. Some of the real-world projects of wireless sensor networks are Bethymetry [77], Ocean Water Monitoring [78], ZebraNet [79], Cattle Herding [80], and Bird Observation on Great Duck Island [81], Grape Monitoring [82], Rescue of Avalanche Victims [83], and neuRFon [84].

In wireless sensor network, nodes are uniformly or stochastically deployed in the target region. Interconnection of nodes in a decentralized fashion without following any preexisting infrastructure is named as AdHoc network. In wireless sensor network, nodes are scattered randomly across the region for decentralizing the network traffic and computational load to increase the coverage, capacity, and reliability. This distributed network scenario also comes up with avoidance of single point of failure. As every node in WSN either acts as sensor, router, or gateway node which has its own computing responsibilities resulting in distributed computing paradigm, uniform or stochastic deployment of wireless sensor nodes comes up with a distributed architecture of nondirected graph where nodes are vertices and the edges represent the communication link between them. In the distributed network of sensor nodes, each computational autonomous entity (node) acts/can act as a sensor as well as router.

This entity establishes communication links with other node in its footprint through omnidirectional full duplex antenna. This cooperative interaction of neighboring nodes expands up to the whole network giving birth to its self-configuring, self-organizing, self-adaptive, and self-synchronizing capabilities. These are extensively exploited by routing algorithms for catering the dynamicity of mobile ad hoc sensor nodes. In computational complexity theory, optimization problem of routing is a special type of computational problem which is concerned with the issues of repetition of algorithmic process until an optimal solution is reached or the required target is met. As the very nature of routing is NP complete (nondeterministic complete) in its optimal solutions search, so resolving such problems in polynomial time is nondeterministic and hence the nature of their solutions is heuristic as such problems cannot always have the definite required solutions. From the case of meta-heuristic solutions of Ant Colony Optimization for its suitability to Mobile Ad hoc Network (MANET), Mehfuz and Doja [85] have described the following reasons; that is why ACO meta-heuristic could perform well in multi-hop scenario of MANET.

(i) *Dynamic Topology.* ACO meta-heuristic is based on agent systems and works with individual ants.

(ii) *Local Work.* ACO meta-heuristic is based on local information that is, no routing table and other information, blocks have to be transmitted to neighbors.

(iii) *Link Quality*. Agent movement updates the link and its quality.

(iv) *Support for Multipath.* Due to pheromone concentration value of neighbor nodes at some specific nodes, this also supports multipath.

Apart from the emergence of bio-inspired solutions in artificial systems in general as is intuited from Table 3, it has a lot of contributions in the intelligent optimization of variety of applications in wireless communication domain specifically. Table 3 shows the list of inspired systems originated from the natural system and their application in WSN domain. References are also cited for the further exploration of interested topics. Without exaggeration, it can be concluded that there is no field, which is left uninspired

by natural behavior. In contrary to this emergence of bio-inspired solutions, Abbott [86] have pointed out the failure of bio-inspired computing to mirror the biological realities. He concluded that these failures arise out of our inability as yet to fully understand what is meant by emergence. But the fact is that bio-inspired computing has the remedies for most of the problems in non-biological systems. As a proponent of this concept, he concluded his article with the claim that bio-inspired computing should be 99% inspiration combined with 1% mimicry. This is the inverse of Thomas Alva Edison's point of view that described invention as 1% inspiration and 99% perspiration. So far as the biological inspiration in the field of network routing esp. mobile wireless sensor network is concerned, there is well established history of this relation in different aspects [87–89].

7. Future Directions

Interdisciplinary research is playing a splendid contribution for various problems' solving. Even the fields those are ever, not only seemed to be very angry to each other but also considered to be on rivalry, have now become indispensable to each other. Computational biology and econometrics are some of those examples. Within the nature, intradisciplinary research of merging the different biological systems is coming up on the scene to further improve the solutions presented earlier by single biological systems. Ideas from cellular automata are merging to genetic algorithms to present new and improved solution to variety of problems [90]. This synergistic mating of biological systems is further needed to figure out the consequences of individual artificial system concocted with other artificial system. This results in improving the algorithms in many dimensions and its applications in various disciplines thus forming ensemble systems.

The design of ensembles can be viewed as a three-stage process: generation of a pool of candidates, selection of some of them from this pool, and combination of the individual outputs to produce a single output that is, a unique solution to a complex problem [91]. This three-stage ensemble process is shown in Figure 3. Although there are no hard and fast rules to make biological hybrid solutions for any type of non-biological problem, the merger of various techniques depends upon the targeted problem and the way the researcher wants to solve it.

Moreover, the effectiveness of the biological solution to solve various artificial problems depends upon the type and scale of the problem.

So to present even a summarized view depicting the relationship between various biological systems and non-biological problems may require more numbers of volumes. Routing issue is solved best by SI as it deals with adaptive, robust, scalable, and distributive properties. A variety of algorithms are involved in SI, that is, fish schooling, bee colony, and ACO algorithms. ANN, GA, and PSO are centralized solutions. ANN and GA have comparable longer run time. The hybrid approach, that is, using multi-classes of CI complement more for high level of optimization which is a future path way for the researcher's as the hybrid approach is lean manufacturing idea, convinced the cerebral drama to have a look to the other side of the picture for better optimization. FL is deterministic and fast as compared to other algorithms but optimization is an issue in it. AIS deals with very little solutions of WSN as it is a better tool than GA but computationally requires more run time. Hence, it is intuited from the above discussion that if a technique is strong in one aspect then on the other hand its other aspect is weak. To make the life easy, we are presenting the core characteristics of some of the biological systems, those come under the umbrella of CI. From the given tabulated view of evaluating parameters and the efficiency of biological techniques against those parameters in Table 4, the researcher can select the appropriate technique or make a hybrid technique out of it to better solve the non-biological problem. For this implication, the nature and characteristics of targeted problem should be carefully identified to get them solved by the selected individual or hybrid biological technique. State variables, not of search points, running time, target problem, and features are efficient to compare and analyze CI paradigm algorithms to solve WSN issues. It can also be applied to a variety of fields to optimize and solve the issues of different application fields of CI. These are all helping to optimize a certain problem. Table 4 shows a comparative analysis of each CI paradigms algorithms. A detailed discussion and description of the evaluation parameters mentioned in it is given in [94]. It also shows that these all will help the researchers to optimize any issue in any field.

Under the artificial systems perceptive, in Figure 4 there are few systems which are enhancing each other's specifications and applications to enhance the traits of one artificial system by using the traits of another artificial system. We can simplify more problems by admiring the traits of artificial systems of this computational modern era. The hybrid algorithms from the intradisciplinary domain of computational intelligence are also a very useful approach and gaining fame for solving the computational problems by integrating the two or more algorithms of this paradigm. The crux of advantages gaining from this merger conclude in a point that the hybrid algorithm formed is capable to eliminate the weakness of one algorithm and polishes the benefits of other algorithms for solving the relevant issue which cannot be solved more efficiently with one algorithm of this solution domain.

The examples of such hybrid computational intelligence algorithms are evolutionary swarm intelligence, Swarm-based Fuzzy Logic Control, Genetic Algorithm-Neural network, genetic-fuzzy intelligence, neuroFuzzy Technique, PSO-Genetic Algorithm, and Neuro-Immune Systems. These hybrid systems are also used in wireless sensor networks domain.

Table 5 shows few of the issues of WSN which are solved by hybrid algorithms of CI. The Neuro fuzzy algorithm [68] optimizes the clustering issue by passing the parameters of each sensor nodes such as memory, available power, and processing speed of each sensor nodes. These inputs are mapped to the fuzzy logic's set of rules to achieve functional objectives, so that the output actions are verified

Figure 3: Three stage process of ensembles design.

and optimized against the input information. The output of fuzzier is monitoring coefficients for each node. It depends on power availability, memory availability, and processing speed of sensor node. Finally, input the monitoring coefficients to the neural network to obtain clustering depending on the application requirement.

Similarly Yanjing and Li [67] optimize the learning issue in WSN and combine the ability of evolution of Genetic Algorithm and probability searching of Swarm Intelligence. In the future, we can develop more hybrid algorithms for optimizing the solutions of memory and computation power which alternatively affect other issues of wireless sensor networks and Table 5 can be a much help in it. More effective approaches of hybrid algorithms of CI paradigm can be used to solve the issues of WSN more efficiently. Few of them are shown in Figure 4. Apart from the emergence of bio-inspired solutions in artificial systems in general as is intuited form Table 3, it has a lot contributions in the intelligent optimization of variety of applications in wireless communication domain specifically.

8. Conclusion

In this paper, we have elaborated the importance of bio-inspired algorithms for the optimal solutions of non-biological systems. We have figured out the key constituents of biological systems and their exploitation in developing the bio-inspired systems. It has been intuited from the Table 2 that the said solutions have their emergent importance in the intelligent optimization in a variety of applications in wireless

Table 4: Comparative analysis of various CI techniques.

Evaluation parameters	Artificial neural network (ANN)	Artificial immune system (AIS)	Fuzzy logic (FL)	Genetic algorithm (GA)	Swarm intelligence (SI)
Development epoch	*1969*	*1986*	*1987*	*Late 1960's*	*Early 1990's*
State variable	Mixed variable	Continuous and discrete variables	Discrete variable	Discrete variable	Continuous, discrete, and mixed variables
No of search points	*Multipoint search*	*Multipoint search*	*Multipoint search*	*Multipoint search*	*Multipoint search*
Solution guarantee	Rarely offers entire solution	Best for time varying solution	Appropriate	Definite in favorable ways	Precise
Run time	*Long*	*Medium*	*Short*	*Medium*	*Medium*
Target problem	- Combinatorial optimization - Multiobjective optimization	- Combinatorial optimization - Multiobjective optimization - Continuous optimization	- Combinatorial optimization - Multiobjective optimization	- Combinatorial optimization - Global optimization - Nonlinear optimization	- Combinatorial optimization - Continuous optimization - Nonlinear optimization
Features	*- Adaptive learnin - Self-organization - Fault tolerant*	*- Provides tools in local and global search - Powerful optimization tool - Self-adaptive and self-learning*	*- Improve system nonlinearity - Deterministic - Performs best even with small input - Accurate*	*- Solutions to optimization and search problems - Good global solutions*	*- Distributive approach - Self-organization - Decentralized control - Powerful optimization tool*

Table 5: Hybrid bio-inspired-technique-based proposed solutions for various issues in wireless sensor ad hoc networks.

Hybrid Bio-inspired Technique	Proposed solution for various issues in wireless sensor ad hoc network
Neuroimmune	*Jabbari and Lang [63] have targeted the quality of service; network records are evaluated using an efficient data processing algorithm comprising of Neural Network and Artificial Immune System.*
Immune-swarm	Saleem et al. [64] have proposed nature inspired autonomous mechanism for network security without any central control.
Neurogenetic swarm	*Mishra and Patra [65] have proposed a hybrid technique that encompasses three bio-inspired techniques; Neural Network, Genetic Algorithm, and Particle Swarm Optimization to short-term forecasting of the load on nodes in sensor networks.*
Swarm fuzzy	Cui et al. [66] have presented a technique for node identification in the network without the help of global positioning system. This algorithm specifically targets the unsafe contaminant locations like border areas.
Genetic swarm	*Yanjing and Li [67] targets the design and deployment aspects of wireless sensor network. His proposed hybrid technique; Evolutionary and Particle Swarm Optimization Algorithms, ensures the maximum coverage of the underlying area by the sensor nodes.*
Neurofuzzy	The proposed cluster designing technique by Veena and Vijaya Kumar [68] is a merger of Neural Network and Fuzzy Logic bio-inspired techniques. The algorithm targets the routing of aggregated data in mobile wireless sensor nodes where the designed clusters are in dynamic form.
Genetic fuzzy	*Lui et al. [69] have given an idea for routing in wireless ad hoc networks. He proposed multipath routing technique that adopts its path according to the changing network topology in efficient way with respect to energy.*
Genetic neural	Gao and Tian [70] have targeted wireless sensor network security issue. He proposed that the data acquisition node uses sensors to collect information and processes them by image detection algorithm and then transmits information to control centre with wireless mesh network.The ability of strong self-learning and pattern classification and fast convergence of genetic-neural system resolves the defect and improves the intelligence.

communication domain. This paper has well elucidated the fact that real creation of the One has really the marvelous traits in it. These traits act as a guiding light for achieving the optimal solution of any artificial system. Biological systems ever provide the better solutions through its intelligent optimization techniques. If there arises any gap between the bio-inspired solution and the artificial systems' problem then this gap can be bridged by carefully identifying the behavioral and architectural traits to develop the metaphoric relationship. The upshot of this paper leads to one step ahead to intradisciplinary research within the biological systems to come up with better bio-inspired solutions. This paper leads

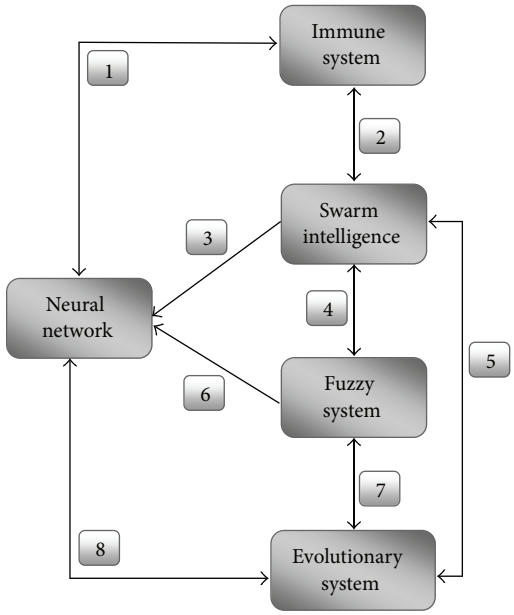


Figure 4: Hybrid bio-inspired techniques.

us to differentiate between various optimization problems existing in WSN collectively showing that hybrid and non-hybrid algorithms can efficiently optimize the problems in wireless sensor networks. The hybrid algorithms open new horizon in the optimization solutions of WSN.

Acknowledgments

The authors are very thankful to Mr. Abdullah, Mr. ShafAlam, Ms. Sania Tariq, and Mr. Muhammad Irfan for their useful discussion on this research work.

References

[1] 2011, http://www.wonderfulinfo.com/winfo/muslminv.php.

[2] 2012, http://articles.cnn.com/2010-01-29/world/muslim.invention_1_hassani-inventions-muslim/3?_s=PM:WORLD.

[3] S. Lin and B. W. Kernighan, "An elective heuristic algorithm for the travelling salesman problem," *Journal of Operations Research*, vol. 21, pp. 498–516, 1973.

[4] R. Dorne and J. Hao, "Tabu search for graph coloring, T-colorings and set T-colorings," in *Meta-Heuristics: Advances and Trends in Local Search Paradigms For Optimization*, pp. 77–92, Kluwer Academic Publishers, Boston, Mass, USA, 1999.

[5] R. Schoonderwoerd, J. L. Bruten, O. E. Holland, and L. J. M. Rothkrantz, "Ant-based load balancing in telecommunications networks," *Adaptive Behavior*, vol. 5, no. 2, pp. 169–207, 1996.

[6] F. Glover, "Tabu search-Part I," *ORSA Journal on Computing*, vol. 1, no. 3, pp. 190–206, 1989.

[7] F. Glover, "Tabu search-Part II," *ORSA Journal on Computing*, vol. 2, no. 1, pp. 4–32, 1990.

[8] F. Glover and M. Laguna, *Tabu Search*, Kluwer Academic Publishers, Boston, Mass, USA, 1997.

[9] V. Cerny, "A thermodynamical approach to the traveling salesman problem," *Journal of Optimization Theory and Applications*, vol. 45, no. 1, pp. 41–51, 1985.

[10] S. Kirkpatrick, C. Gelatt Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Journal of Science*, vol. 220, pp. 671–680, 1983.

[11] H. R. Lourenco, O. Martin, and T. Stutzle, "Iterated local search," in *Handbook of Metaheuristics of International Series in Operations Research & Management Science*, F. Glover and G. Kochenberger, Eds., pp. 321–353, Kluwer Academic Publishers, Norwell, Mass, USA, 2002.

[12] L. Fogel, A. J. Owens, and M. J. Walsh, *Artificial Intelligence Through Simulated Evolution*, John Wiley & Sons, New York, NY, USA, 1966.

[13] J. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, Mich, USA, 1975.

[14] I. Rechenberg, *Evolutionsstrategie: Optimierung Technischer Systemenach Prinzipien Der Biologischen Information*, Fromman, Freiburg, Germany, 1973.

[15] H. P. Schwefel, *Numerical Optimization of Computer Models*, John Wiley & Sons, Chichester, UK, 1981.

[16] M. Dorigo and L. M. Gambardella, "Ant colonies for the travelling salesman problem," *BioSystems*, vol. 43, no. 2, pp. 73–81, 1997.

[17] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, 1997.

[18] L. M. Gambardella and M. Dorigo, "An ant colony system hybridized with a new local search for the sequential ordering problem," *INFORMS Journal on Computing*, vol. 12, no. 3, pp. 237–255, 2000.

[19] M. Dorigo and T. Stützle, "The ant colony optimization metaheuristic: algorithms, applications and advances," in *Handbook of Metaheuristics*, F. Glover and G. Kochenberger, Eds., vol. 57 of *International Series in operations Research & Management Science*, pp. 251–285, Kluwer Academic Publishers, Norwell, Mass, USA, 2000.

[20] M. Marks, "A survey of multi-objective deployment in wireless sensor networks," *Jounal of Telecommunication and Information Technology*, no. 3, pp. 36–41, 2010.

[21] X. Wei and L. Zhi, "The multi-objective routing optimization of WSNs based on an improved ant colony algorithm," in *Proceedings of the 6th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '10)*, September 2010.

[22] J. Mao, Z. Wu, and X. Wu, "A TDMA scheduling scheme for many-to-one communications in wireless sensor networks," *Computer Communications*, vol. 30, no. 4, pp. 863–872, 2007.

[23] L. Tie and J. Li, "Combinatorial optimization for wireless sensor networks," in *Proceedings of the IEEE International Workshop on VLSI Design and Video Technology*, pp. 28–30, Suzhou, China, May 2005.

[24] R. V. Kulkarni, G. K. Venayagamoorthy, and M. X. Cheng, "Bio-inspired node localization in wireless sensor networks," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '09)*, pp. 205–210, October 2009.

[25] J. Wang, H. He, B. Chen, Y. Chen, and T. Guan, "Data aggregation and routing in wireless sensor networks using improved ant colony algorithm," in *Proceedings of the International Forum on Computer Science-Technology and Applications (IFCSTA '09)*, pp. 215–218, December 2009.

[26] C. Lee, *On Quality of Service Management*, School Of Computer Science, Carnegie Mellon University, Pittsburgh, Pa, USA, 1999.

[27] R. A. Johnson, "Learning, memory, and foraging efficiency in two species of desert seed-harvester ants," *Ecology*, vol. 72, no. 4, pp. 1408–1419, 1991.

[28] W.-L. Chang, M. Guo, and M. S.-H. Ho, "Fast parallel molecular algorithms for DNA-based computation: factoring integers," *IEEE Transactions on Nano-Bioscience*, vol. 4, no. 2, pp. 149–1163, 2005.

[29] G. Păun, *Introduction to Membrane Computing*, Institute of Mathematics of the Romanian Academy, Research Group on Natural Computing Department of Computer Science and Artificial Intelligence University of Sevilla, 2006.

[30] S. Wolfram, "Statistical mechanics of cellular automata," *Reviews of Modern Physics*, vol. 55, no. 3, pp. 601–644, 1983.

[31] X. Wei and L. Zhi, "The multi-objective routing optimization of WSNs based on an improved ant colony algorithm," in *Proceedings of the 6th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '10)*, September 2010.

[32] J. Dollner and K. Hinrichs, "A generic rendering system," *IEEE Transactions of Visualization and Computer Graphics*, vol. 8, no. 2, pp. 99–118, 2002.

[33] H. He, Z. Zhu, and E. Mäkinen, "A neural network model to minimize the connected dominating set for self-configuration of wireless sensor networks," *IEEE Transactions on Neural Networks*, vol. 20, no. 6, pp. 973–982, 2009.

[34] A. Chernihovskyi, C. E. Elger, and K. Lehnertztl, "Effect of in hbitory diffusive coup ling on frequency-selectivity of excitable media stimulated with cellular Neural Networks," in *Proceedings of the 10th International Workshop on Cellular Neural Networks and Their Applications*, pp. 28–30, Istanbul, Turkey, August 2006.

[35] A. K. Jain, J. Mao, and K. M. Mohiuddin, "Artificial neural networks: a tutorial," *IEEE Transactions on Computer*, vol. 29, no. 3, pp. 31–44, 1996.

[36] L. N. de Castro and J. Timmis, *Artificial Immune Systems: A Computational Approach*, Springer, 2002.

[37] Encylopedia Article, "Genetic algorithm," Access Science from McGraw-Hill.

[38] W. Barker, D. M. Halliday, Y. Thoma, E. Sanchez, G. Tempesti, and A. M. Tyrrell, "Fault tolerance using dynamic reconfiguration on the POEtic tissue," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 5, pp. 666–684, 2007.

[39] C. M. Ionescu, I. Muntean, J. A. Tenreiro-Machado, R. De Keyser, and M. Abrudean, "A theoretical study on modeling the respiratory tract with ladder networks by means of intrinsic fractal geometry," *IEEE Transactions on Bio-Medical Engineering*, vol. 57, no. 2, pp. 246–253, 2010.

[40] G. Rozenberg and A. Salomaa, *The Mathematical Theory of L Systems*, Academic Press, New York, NY, USA, 1980.

[41] A. M. Tyrrell, P. C. Haddow, and J. Torresen, *Evolvable Systems: From Biology to Hardware: 5th International Conference*, 2003.

[42] 2012, http://dictionary.reference.com/browse/artificial%20life.

[43] W. Leventon, "Synthetic skin," *IEEE Spectrum*, vol. 39, no. 12, pp. 28–33, 2002.

[44] J. Podpora, L. Reznik, and G. Von Pless, "Intelligent real-time adaptation for power efficiency in sensor networks," *IEEE Sensors Journal*, vol. 8, no. 12, pp. 2066–2073, 2008.

[45] A. I. Moustapha and R. R. Selmic, "Wireless sensor network modeling using modified recurrent neural networks: application to fault detection," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 5, pp. 981–988, 2008.

[46] H. He, Z. Zhu, and E. Mäkinen, "A neural network model to minimize the connected dominating set for self-configuration of wireless sensor networks," *IEEE Transactions on Neural Networks*, vol. 20, no. 6, pp. 973–982, 2009.

[47] 2012, http://www.niaid.nih.gov/topics/immuneSystem/Pages/whatIsImmuneSystem.aspx.

[48] S. Sarafijanović and J. Y. Le Boudec, "An artificial immune system approach with secondary response for misbehavior detection in mobile ad hoc network," *IEEE Transactions on Neural Networks*, vol. 16, no. 5, pp. 1076–1087, 2005.

[49] H. Wang, D. Peng, W. Wang et al., "Artificial immune system based image pattern recognition in energy efficient wireless multimedia sensor networks," in *Proceedings of the IEEE Military Communications Conference (MILCOM '08)*, November 2008.

[50] S. Yang, H. Cheng, and F. Wang, "Genetic algorithms with immigrants and memory schemes for dynamic shortest path routing problems in mobile ad hoc networks," *IEEE Transactions on Systems, Man and Cybernetics Part C*, vol. 40, no. 1, pp. 52–63, 2010.

[51] X.-M. Hu, J. Zhang, Y. Yu et al., "Hybrid genetic algorithm using a forward encoding scheme for lifetime maximization of wireless sensor networks," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 5, pp. 766–781, 2010.

[52] R. O. Cunha, A. P. Silva, A. A. F. Loreiro, and L. B. Ruiz, "Simulating large wireless sensor networks using cellular automata," in *Proceedings of the 38th Annual Simulation Symposium (ANSS '05)*, pp. 323–330, April 2005.

[53] S. Wang and H. Wu, "An improved coverage scheme based on cellular automata in WSN," in *Proceedings of the 2nd International Conference on Networks Security, Wireless Communications and Trusted Computing (NSWCTC '10)*, pp. 458–461, April 2010.

[54] M. Li and Y. Liu, "Rendered path: range-free localization in anisotropic sensor networks with holes," *IEEE/ACM Transactions on Networking*, vol. 18, no. 1, pp. 320–332, 2010.

[55] M. Li and Y. Liu, "Rendered path: range-free localization in anisotropic sensor networks with holes," *IEEE/ACM Transactions on Networking*, vol. 18, no. 1, pp. 320–332, 2010.

[56] J. T. Huang, J. H. Shiao, and J. M. Wu, "A miniaturized Hilbert inverted-F antenna for wireless sensor network applications," *IEEE Transactions on Antennas and Propagation*, vol. 58, no. 9, pp. 3100–3103, 2010.

[57] E. Felemban, S. Vural, R. Murawski et al., "SAMAC: a cross-layer communication protocol for sensor networks with sectored antennas," *IEEE Transactions on Mobile Computing*, vol. 9, no. 8, pp. 1072–1088, 2010.

[58] M. C. Vuran and I. F. Akyildiz, "XLP: a cross-layer protocol for efficient communication in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 11, pp. 1578–1591, 2010.

[59] D. Pompili and I. F. Akyildiz, "Overview of networking protocols for underwater wireless communications," *IEEE Communications Magazine*, vol. 47, no. 1, pp. 97–102, 2009.

[60] L. M. Fernandez-Carrasco, H. Terashima-Marin, and M. Valenzuela-Rendon, "On the possibility to design intelligent sensor systems after excitable media models: an agent-based simulation," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC 2008)*, pp. 1181–1186.

[61] H. He, Z. Zhu, and E. Mäkinen, "A neural network model to minimize the connected dominating set for self-configuration of wireless sensor networks," *IEEE Transactions on Neural Networks*, vol. 20, no. 6, pp. 973–982, 2009.

[62] S. Aeron, V. Saligrama, and D. A. Castanon, "Efficient sensor management policies for distributed target tracking in multihop sensor networks," *IEEE Transactions on Signal Processing*, vol. 56, no. 6, pp. 2562–2574, 2008.

[63] A. Jabbari and W. Lang, "Advanced bio-inspired plausibility checking in a wireless sensor network using Neuro-immune systems: autonomous fault diagnosis in an intelligent transportation system," in *Proceedings of the 4th International Conference on Sensor Technologies and Applications (SENSORCOMM '10)*, pp. 108–114, July 2010.

[64] K. Saleem, N. Fisal, M. S. Abdullah, A. B. Zulkarmwan, S. Hafizah, and S. Kamilah, "Proposed nature inspired self-organized secure autonomous mechanism for WSNs," in *Proceedings of the 1st Asian Conference on Intelligent Information and Database Systems (ACIIDS '09)*, pp. 277–282, April 2009.

[65] S. Mishra and S. K. Patra, "Short term load forecasting using neural network trained with genetic algorithm & particle swarm optimization," in *Proceedings of the 1st International Conference on Emerging Trends in Engineering and Technology (ICETET '08)*, pp. 606–611, July 2008.

[66] X. Cui, T. Hardin, R. K. Ragade, and A. S. Elmaghraby, "A swarm-based fuzzy logic control mobile sensor network for hazardous contaminants localization," in *Proceedings of the IEEE International Conference on Mobile Ad-Hoc and Sensor Systems*, pp. 194–203, October 2004.

[67] S. Yanjing and L. Li, "Hybrid learning algorithm for effective coverage in wireless sensor networks," in *Proceedings of the 4th International Conference on Natural Computation (ICNC '08)*, pp. 227–231, October 2008.

[68] K. N. Veena and B. P. Vijaya Kumar, "Dynamic clustering for Wireless Sensor Networks: a neuro-fuzzy technique approach," in *Proceedings of the IEEE International Conference on Computational Intelligence and Computing Research (ICCIC '10)*, pp. 151–156, December 2010.

[69] H. Liu, J. Li, Y. Q. Zhang, and Y. Pan, "An adaptive genetic fuzzy multi-path routing protocol for wireless ad-hoc networks," in *Proceedings of the 6th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and 1st ACIS International Workshop on Self-Assembling Wireless Networks, SNPD/SAWN 2005*, pp. 468–475, May 2005.

[70] M. Gao and J. Tian, "Wireless sensor network for community intrusion detection system based on improved genetic algorithm neural network," in *Proceedings of the International Conference on Industrial and Information Systems (IIS '09)*, pp. 199–202, April 2009.

[71] J. Brownlee, "On biologically inspired computation, a.k.a. The field," Technical Report 5-02, 2005.

[72] K. N. Lodding and A. Nas, "The hitchhiker's guide to biomorphic software," *Queue*, vol. 2, no. 4, pp. 66–75, 2004.

[73] P. Marrow, "Nature-inspired computing technology and applications," *BT Technology Journal*, vol. 18, no. 4, pp. 13–23, 2000.

[74] R. Iram, S. Jabbar, and A. A. Minhas, "Developing biological metaphor in Ad-Hoc Networks," . In press.

[75] K. Martinez, R. Ong, J. K. Hart, and J. Stefanov, "Glacsweb: a sensor web for glaciers," in *Proceedings of the IEEE 1st European Workshop on Wireless Sensor Networks (EWSN '04)*, Berlin, Germany, January 2004.

[76] F. Michahelles, P. Matter, A. Schmidt, and B. Schiele, "Applying wearable sensors to avalanche rescue," *Computers and Graphics*, vol. 27, no. 6, pp. 839–847, 2003.

[77] K. Rlomer and F. Mattern, "The design space of wireless sensor networks," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 54–61, 2004.

[78] W. Marshall, C. Roadknight, I. Wokoma, and L. Sacks, *Self-Organizing Sensor Networks*, " UbiNet, London, UK, 2003.

[79] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet," in *Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems*, pp. 96–107, San Jose, Calif, USA, October 2002.

[80] Z. Butler, P. Corke, R. Peterson, and D. Rus, "Networked cows: virtual fences for controlling cows," in *Workshop on Applications of Mobile Embedded Systems (WAMES '04)*, Boston, Mass, USA, June 2004.

[81] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 88–97, Atlanta, Ga, USA, September 2002.

[82] R. Beckwith, D. Teibel, and P. Bowen, "Pervasive computing and proactive agriculture," in *Proceedings of Second International Conference of Pervasive Computing*, Vienna, Austria, April 2004.

[83] F. Michahelles, P. Matter, A. Schmidt, and B. Schiele, "Applying wearable sensors to avalanche rescue," *Computers and Graphics*, vol. 27, no. 6, pp. 839–847, 2003.

[84] Neurfon, 2012, http://www.motorola.com/content.jsp?globalObjectId=290.

[85] S. Mehfuz and M. N. Doja, "Swarm intelligent power-aware detection of unauthorized and compromised nodes in MANETs," *Journal of Artificial Evolution and Applications*, vol. 2008, Article ID 236803, 16 pages, 2008.

[86] R. Abbott, "Challenges for biologically-inspired computing," in *Proceedings of the Workshops on Genetic and Evolutionary Computation (GECCO '05)*, pp. 12–22, Washington, DC, USA, June 2005.

[87] G. Di Caro, F. Ducatelle, and L. Maria Gambardella, "AntHocNet: an adaptive nature-inspired algorithm for routing in mobile Ad hoc networks," Technical Report IDSIA-27-04-2004, 2004.

[88] M. Gunes, U. Sorges, and I. Bouazizi, "ARA—the ant-colony based routing algorithm for MANETs," in *Proceedings of the ICPP International Workshop on Ad Hoc Networks (IWAHN '02)*, pp. 79–85, Vancouver, Canada, August 2002.

[89] M. Roth and S. Wicker, "Termite: Ad-Hoc Networking with Stigmergy," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '03)*, vol. 5, pp. 2937–2941, December 2003.

[90] E. Sapin, L. Bull, and A. Adamatzky, "Genetic approaches to search for computing patterns in cellular automata," *IEEE Computational Intelligence Magazine*, vol. 4, no. 3, pp. 20–28, 2009.

[91] P. A. D. Castro and F. J. Von Zuben, "Learning ensembles of neural networks by means of a Bayesian artificial immune system," *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 304–316, 2011.

[92] N. Y. Nikolaev and H. Iba, "Learning polynomial feed forward neural networks by genetic programming and back propagation," *IEEE Transactions on Neural Networks*, vol. 14, no. 2, pp. 337–350, 2003.

[93] F. Chen, G. Chen, G. He, X. Xu, and Q. He, "Universal perceptron and DNA-like learning algorithm for binary neural networks: LSBF and PBF implementations," *IEEE Transactions on Neural Networks*, vol. 20, no. 10, pp. 1645–1658, 2009.

[94] R. Iram, M. I. Sheikh, S. Jabbar, and A. A. Minhas, "Computational intelligence based optimization of energy aware routing in WSN," in *Proceedings of the International Conference on Soft Computing and Applications (ICSCA'11)*, San Francisco, Calif, USA, October 2011.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 570805, 15 pages
http://dx.doi.org/10.1155/2013/570805

Research Article

Energy-Balanced Separating Algorithm for Cluster-Based Data Aggregation in Wireless Sensor Networks

Shuai Fu,[1] Jianfeng Ma,[1] Hongtao Li,[1] and Changguang Wang[2]

[1] *School of Computer Science and Technology, Xidian University, Xi'an 710071, China*
[2] *College of Information Technology, Hebei Normal University, Shijiazhuang 050016, China*

Correspondence should be addressed to Shuai Fu; ljhfs0803@126.com

Received 23 September 2012; Revised 16 December 2012; Accepted 22 December 2012

Academic Editor: Jiman Hong

Clustering provides an effective way to prolong the lifetime of wireless sensor networks. However, the head node may die much faster than other nodes due to its overburden. In this paper, we design a method to mitigate the uneven energy dissipation problem. Considering the relaying load undertaken by each cluster, we use the network topology and energy consumption to calculate a cluster radius for obtaining the intercluster energy balancing. A new cluster-leader election algorithm is proposed wherein the task of a single cluster head is separated to two nodes so that the critical nodes in each cluster will not exhaust their power so quickly. Furthermore, cross-level data transmission is used to prolong network lifetime. Extensive simulation experiments are carried out to evaluate the method with several performance criteria. Our simulation results show that this method obtains satisfactory performance on balancing energy dissipation and prolonging the networks lifetime.

1. Introduction

Continued advances of microelectromechanical systems (MEMS) and wireless communication technologies have enabled the deployment of large-scale wireless sensor networks (WSNs) [1]. Due to limited and nonrechargeable energy provision of sensors, improving energy efficiency and maximizing the network lifetime by decreasing energy consumption of the individual nodes and balancing energy consumption of all nodes are the major challenges in the research of data aggregation algorithms in WSNs [2].

The energy of a sensor node is mainly consumed by the communication unit, computing unit, and sensor, out of which the wireless transceiver uses a large portion of the energy. The traffic follows a multihop pattern, where intermediate nodes deplete their energy faster when taking more tasks, which leads to what is known as an energy hole [3]. Therefore, unbalanced energy consumption is an inherent problem, which needs to be solved to prolong the network lifetime.

Clustering method for data aggregation in wireless sensor network has attracted great attention for its high efficiency [4–6]. The data traffic (as well as the data transmission and reception energy) can be greatly reduced by applying data aggregation at cluster heads. It significantly reduces the battery drainage of individual sensors and also has other advantages in terms of simplifying network management, improving security, and achieving better scalability. Recently, some studies have been done to address issues related to energy efficiency and prolonging the lifetime [7] of the WSNs. In this work, our main focus is rather on balancing the energy dissipation of the whole network and making energy-efficient routing during data aggregation. It is considered from two aspects: intercluster energy balancing and intracluster energy balancing. The motivation and main contributions of this paper are listed in the following.

An analysis of energy balancing problem is made in WSNs under cluster hierarchy. This problem deals with both intercluster and intracluster. As to the former, we try to allow each cluster to consume approximately the same amount of energy through arranging cluster sizes. As for the latter, we design an algorithm from the task separation perspective. Through this approach, the load imposed on a single cluster head can be alleviated in each cluster. Although a lot of literatures on dividing the network into clusters cope with the problem of unbalanced power consumption in WSNs,

none of the existing algorithms consider assigning the tasks of CHs to two nodes for intracluster energy balancing. The main contributions of this paper are summarized as follows.

(i) Arranging cluster sizes based on the equal intercluster energy consumption. According to the relaying load of each cluster, the cluster radius is calculated. We only consider the energy consumption on data aggregation and transmission. It is assumed that the total energy consumed by each cluster is approximately the same. As leaders of clusters near the BS will relay more data than those located far away from the BS, their radiuses will be smaller accordingly.

(ii) Designing the intracluster communication algorithm from the task separation perspective. To slow the energy consumption of critical nodes in each cluster, the algorithm is designed with consideration of task separation. Both the data gathering and aggregation are performed by a sensor named processor, and the report to the base station will be done by another sensor named forwarder in the same cluster. The election procedure of the processor and the forwarder is performed simultaneously, thus avoiding wasting the bandwidth caused by transmitting messages too many times.

The remainder of this paper is organized as follows. Section 2 summarizes related work. Section 3 describes the network model and elaborates the imbalanced energy consumption problem that we address in this work. In Section 4, the proposed method for arranging cluster size is described in detail. Section 5 shows the new algorithm for cluster-leader election. Section 6 gives a performance analysis of the proposed algorithm. We make a theoretical energy consumption analysis in Section 7. Then in Section 8, we evaluate the performance of our approach by simulation and make a comparison of it with LEACH, MR-LEACH, EECA, and ACT. Finally, we conclude the paper in Section 9.

2. Related Work

In this section, four steps of the hierarchical routing protocol and related works are introduced: CH election, cluster formation, intracluster communications, and intercluster communications.

In CH election, many typical protocols adopt different approaches. The first proposed cluster-based algorithm for WSNs is LEACH [8]. It divides the operation into rounds and randomly selects new CHs in each round to distribute the energy load among all nodes. In the data transmission phase, each cluster head forwards an aggregated packet to the base station directly. One common issue with LEACH is that the energy of sensor nodes which are located far away from their CHs can be easily used up for the transmission of packets to their CHs.

Several variants of LEACH protocol are proposed to make an improvement on it that further decreases the power consumption. LEACH-C [8] is a centralized version of LEACH. It uses the BS central control to form clusters. During the set-up phase, each node sends information about its current location and energy level to the BS. Then the BS computes the average node energy and chooses those whose energy level is above this average as candidates for CHs. For minimizing the total sum of squared distances between all the non-CHs and the closest CH, LEACH-C uses the simulated annealing algorithm to find the optimal clusters. Fan and Song [9] introduce the energy-LEACH protocol (E-LEACH), which chooses CHs based on remaining energy. Multihop communication mode among cluster heads is adopted to avoid the whole network from dying quickly and prolong the network lifetime. Loscri et al. [10] build a two-level hierarchy for LEACH (TL-LEACH). TL-LEACH considers a randomized rotation of the CHs and chooses one of the CHs that lies between the current CH and the BS as a relay station. This allows CHs to better distribute the energy load among sensor nodes when the network density is higher. Yassein et al. [11] present the concept of vice-CH (V-LEACH), a sensor node which will become a CH if the current CH uses up its energy. This ensures that cluster nodes data will always reach the BS. Farooq et al. [12] present a multihop routing with low energy adaptive clustering hierarchy (MR-LEACH) protocol. The CH election in MR-LEACH is based on the available energy, and it partitions the network into different layers of clusters. CHs in each layer are responsible for relaying data for CHs at lower layers to transmit data to the BS. Thus, MR-LEACH follows multihop routing from cluster heads to the BS to conserve energy.

During cluster formation, CHs broadcast messages and non-CH nodes determine which cluster to join according to the signal strength received. As all nodes tend to join the closest cluster, clusters are formed in various sizes. The greater the cluster is, the heavier the load of its CH is. An energy-efficient clustering scheme (EECS) [13] presented by Ye et al. takes into account the unbalanced energy dissipation. In EECS, during the cluster formation phase nodes decide to associate with a CH based on a weighted cost factor that is composed of three functions. A new scheme was given to avoid the energy hole problem with unequal clustering mechanism in [14]. Its core is an energy-efficient uneven clustering algorithm (UCR) for network topology organization, in which tentative cluster heads use uneven competition ranges to construct clusters of uneven sizes. Some other works attempt to take measures to adjust the size of each cluster so as to reduce the differences of loads between CHs [15, 16]. Paper [17] proposed a novel cluster-based routing protocol named ACT, which aims to reduce the size of clusters near the base station. It provides a method to arrange cluster size, allowing each CH to consume approximately the same amount of energy. However, the CHs are determined as soon as the cluster radius is obtained. Their locations are closest to the ideal but may not be the best.

As for intracluster communications, some studies suggest that the sleep mode of sensor nodes should be adopted in intracluster communications to save energy. That means there is only one node or several nodes in a cluster that are active while the others enter sleep mode (e.g., cluster members take turns collecting data). However, scheduling sleep time is a major issue worthy of discussion [18].

During intercluster communications, the farther the messages to be transmitted, the greater the energy dissipation will be. In [19], the authors proposed an energy-efficient clustering algorithm (EECA) in which the data aggregation tree is constructed by determining the weight of CHs, but this many-to-one communication mode still possesses the imbalance power dissipation problem. Distributed clustering algorithms were proposed in [20], with the objective of minimizing the energy spent in communicating information to the sink. It should be noted that minimizing the total energy consumption is not equivalent to maximizing coverage time, as the former criterion does not guarantee balanced power consumption at various CHs.

Unlike previous approaches, we try to solve the unbalanced energy consumption problem from the perspective of both intercluster communication and intracluster communication. Arranging cluster radius based on the assumption of equal total energy dissipation ensures the energy balance among clusters, while the new separating cluster-based algorithm (SCA) obtains the energy balance among sensors within a cluster. The separation of the CH role alleviates the burden of a single CH, thus avoiding early network collapse due to the death of critical nodes and prolonging the network lifetime.

3. Preliminaries

3.1. Network Model. In this paper, we consider a sensor network consisting of N sensor nodes uniformly dispersed in the service area of the network whose coverage area is a rectangular region of $L \times W$. We make some assumptions about the sensor nodes and the underlying network model.

(1) The positions of BS and sensor nodes are fixed. Nodes are uniformly distributed in the sensor field with density ρ.

(2) Each node is assigned a unique identifier (ID). Sensors are with the same initial energy and their transmit power is controllable. The maximum power level can be used in transmitting data to BS directly.

(3) Links are symmetric. A node can compute the approximate distance to another node based on the received signal strength, if the transmitting power is known.

(4) Sensor nodes can recognize their geographical position and the BS's position via exchanging information.

(5) All sensors are sensing the environment at the same fixed rate and thus always have data to send to the end-user. The size of each data packet is the same.

We use the typical energy consumption model [8]. The energy spent for transmitting an l-bit message over distance d is

$$E_{\mathrm{TX}}(l,d) = \begin{cases} l \times E_{\mathrm{elec}} + l \times \varepsilon_{\mathrm{fs}} \times d^2, & d \prec d_0, \\ l \times E_{\mathrm{elec}} + l \times \varepsilon_{\mathrm{amp}} \times d^4, & d \geq d_0, \end{cases} \quad (1)$$

where E_{elec} is the energy dissipated per bit to run the transmitter or the receiver circuit, $\varepsilon_{\mathrm{fs}}$ and $\varepsilon_{\mathrm{amp}}$ are the energy dissipated per bit to run the transmit amplifier depending on the distance between the transmitter and receiver. If the distance is less than a threshold d_0, the free space (fs) model is used; otherwise, the multipath (mp) model is used.

To receive this message, the expended energy is

$$E_{\mathrm{RX}}(l) = l \times E_{\mathrm{elec}}. \quad (2)$$

The consumed energy of aggregating a message with l-bit is

$$E_A(m,l) = l \times E_{\mathrm{DA}}, \quad (3)$$

where E_{DA} is the energy dissipated per bit to aggregate message signal.

3.2. Related Definition

(1) We denote the ith sensor by S_i and the corresponding sensor node set $S = \{S_1, S_2, \ldots, S_N\}$, where $|S| = N$. For a random node S_i, make its residual energy E_{ri} and its coordinate $L(X_i, Y_i)$.

(2) The neighboring node set R_{CH} of any node S_m is defined as

$$S_m - R_{\mathrm{CH}} = \left\{ S_n \mid d(S_m, S_n) \leq \theta R_m,\ d(S_n, BS) \prec d(S_m, BS) \right\}, \quad (4)$$

where θ is the minimum integer that lets $S_m - R_{\mathrm{CH}}$ contain at least one item (if there does not exist such a θ, define $S_m - R_{\mathrm{CH}}$ as a null).

(3) Define E_{res}_MAX as the threshold of the residual energy of node S_i. If a node's residual energy is less than E_{res}_MAX, it will give up the competition for processor and forwarder. According to (1)–(3), the value of E_{res}_MAX could be estimated by

$$E_{\mathrm{res}}\_\mathrm{MAX} = \mu \times \left[mlE_{\mathrm{elec}} + (m+1)\, lE_{\mathrm{DA}} + (1-\lambda)(m+1)\, l\left(E_{\mathrm{elec}} + \varepsilon_{\mathrm{fs}} d^2\right) \right], \quad (5)$$

where μ represents the number of times of each turn of data acquisition, λ is the compression ratio of data aggregation, d represents the distance between current processor and its parent node, and m represents the number of neighbor nodes.

3.3. Problem Statement. A fundamental issue in WSN is maximizing the network lifetime subject to a given energy constraint. Notice that the BS is usually located far away from the monitoring area. Previous research has shown that multihop intercluster communication mode is usually desirable because of its power-consumption advantage over direct (CH-to-sink) communication (e.g., [21]). However, the energy hole situation is essentially caused because of the

different loads among nodes when the multihop forwarding mode is adopted in intercluster communication.

Based on the given model, balancing the energy consumption to the maximum is our optimization objective. A complete data collection process involves two steps: collecting data from all sensor nodes and delivering the data to the BS. This problem can be formulated as follows:

$$\begin{aligned} &\min \sum_{i \in C_j} \left[E_{\text{ri}} - \overline{E_{\text{res}}^{j}} \right]^2 \\ &\min \sum_{k \in M} \left[\overline{E_{\text{ri}}(k)} - \overline{E_{\text{res}}} \right]^2, \end{aligned} \tag{6}$$

where E_{ri} is the residual energy of node i in cluster j and $\overline{E_{\text{res}}^{j}}$ represents the average remaining energy of nodes in cluster j. $\overline{E_{\text{ri}}(k)}$ represents the average remaining energy of all nodes in cluster k, and $\overline{E_{\text{res}}}$ is the average remaining energy of all sensor nodes.

The above optimization problem can be solved as two subproblems as follows.

(a) How to balance the energy dissipation among nodes within the same cluster? This is referred to as the problem of intracluster energy consumption balancing.

(b) How to balance energy dissipation among different clusters? This is referred to as the problem of intercluster energy consumption balancing.

It will be described in detail in the following two sections.

To enable readers to more easily understand this paper, Table 1 summarizes the notations used in this paper.

Table 1: Notations.

Symbols	Description
N	The number of sensor nodes
A	The area of the sensing field and $A = L \times W$
ρ	The node's density
E_{elec}	The energy dissipated per bit to run the transmitter or the receiver circuit
ε_{fs}	The energy dissipated per bit to run the transmit amplifier when the distance between the transmitter and receiver is less than a threshold
ε_{amp}	The energy dissipated per bit to run the transmit amplifier when the distance between the transmitter and receiver is greater than a threshold
d_0	The threshold distance determining which model to use to calculate the energy consumption of nodes
E_{DA}	The energy dissipated per bit to aggregate message signal
E_{ri}	The residual energy of a node i
E_{res}_MAX	The threshold of residual energy of a node
λ	The compression ratio of data aggregation
m	The number of neighbor nodes
M	The number of different sizes of clusters
r_j	The radius of cluster j
CP_j	The competition bids of being elected as processor for node S_j
CF_j	The competition bids of being elected as forwarder for node S_j
ΔE	The difference value between residual energy of two different clusters
ΔE_T	The difference value between the energy consumption for sending the same data packets to two different clusters

4. Intercluster Energy Balancing

4.1. Arranging Cluster Radius. In the proposed algorithm, we hope to balance the energy consumption between clusters, and this can be achieved by applying (1) and (3) to calculate the radius of each cluster. It is supposed that the tentative network consists of clusters with M different sizes, and each cluster member passes one bit of data to cluster leaders (see Figure 1). The transmission range is regarded as the distance between the centers of two clusters for simplicity in calculations, except in the 1st level (i.e., $(r_m + r_{m-1})$ in Mth level, $(r_{m-1} + r_{m-2})$ in $(M-1)$th level, and so forth).

We assume that the nodes are deployed in each cluster with density ρ. As each cluster leader in the outermost level (Mth level) does not need considering the relay data, it only takes care of the data transmitted by its own cluster members. Its transmission range is $(r_m + r_{m-1})$, and thus the total energy dissipation of each cluster leader in Mth level is

$$\pi r_m^2 \rho E_{\text{DA}} + (1-\lambda)\pi r_m^2 \rho \left[E_{\text{elec}} + \varepsilon_{\text{fs}} (r_m + r_{m-1})^2 \right], \tag{7}$$

where the first part represents the aggregation energy consumption in the Mth level and the second the transmission energy consumption from the Mth level to the $(M-1)$th level.

However, cluster leaders in the $(M-1)$th level not only process data given by their members, but they also perform data relaying for Mth level. According to its transmission range, the total energy dissipation of cluster leaders in $(M-1)$th level is

$$\begin{aligned} &\pi r_{m-1}^2 \rho E_{\text{DA}} + \left[(1-\lambda)\pi r_{m-1}^2 \rho + (1-\lambda)^2 \pi r_m^2 \rho \right] \\ &\quad \times \left[E_{\text{elec}} + \varepsilon_{\text{fs}} (r_{m-1} + r_{m-2})^2 \right]. \end{aligned} \tag{8}$$

Similarly, each cluster leader in the $(M-2)$th level forwards data generated by its own cluster members while performing data relaying for $(M-1)$th level and Mth level. Then the total energy dissipation of each cluster leader in $(M-2)$th level is

$$\begin{aligned} &\pi r_{m-2}^2 \rho E_{\text{DA}} + \left[(1-\lambda)\pi r_{m-2}^2 \rho + (1-\lambda)^2 \pi r_{m-1}^2 \rho + (1-\lambda)^3 \right. \\ &\quad \left. \times \pi r_m^2 \rho \right] \left[E_{\text{elec}} + \varepsilon_{\text{fs}} (r_{m-2} + r_{m-3})^2 \right]. \end{aligned} \tag{9}$$

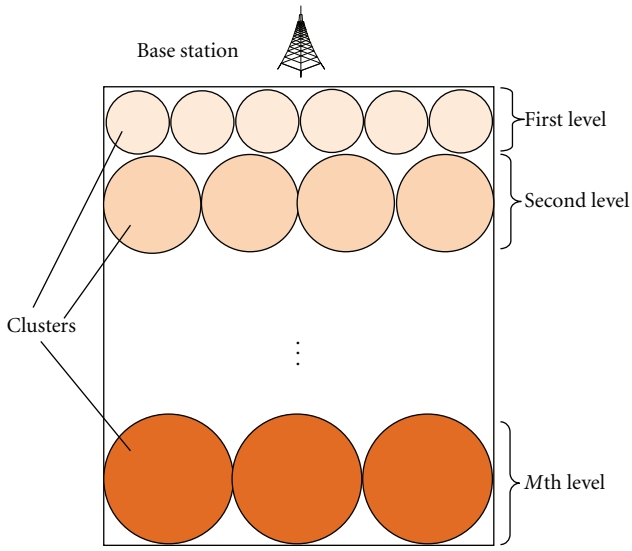


Figure 1: Level model of network topology.

In this way, the total energy dissipation of a cluster leader in each level (for one generated message bit) can be calculated as follows:

$$\begin{aligned}
&\text{The } M\text{th level: } E_m = \pi r_m^2 \rho E_{\mathrm{DA}} + (1-\lambda)\,\pi r_m^2 \rho \\
&\qquad \times \left[E_{\mathrm{elec}} + \varepsilon_{\mathrm{fs}}\left(r_m + r_{m-1}\right)^2\right] \\
&\text{The } (M-1)\text{th level: } E_{m-1} = \pi r_{m-1}^2 \rho E_{\mathrm{DA}} \\
&\qquad + \left[(1-\lambda)\,\pi r_{m-1}^2 \rho \right. \\
&\qquad \left. + (1-\lambda)^2 \pi r_m^2 \rho\right] \\
&\qquad \times \left[E_{\mathrm{elec}} + \varepsilon_{\mathrm{fs}}\left(r_{m-1} + r_{m-2}\right)^2\right] \\
&\qquad \vdots \\
&\text{The 2nd level: } E_2 = \pi r_2^2 \rho E_{\mathrm{DA}} \\
&\qquad + \left[(1-\lambda)\,\pi r_2^2 \rho + (1-\lambda)^2 \pi r_3^2 \rho + \cdots \right. \\
&\qquad \left. + (1-\lambda)^{m-1} \pi r_m^2 \rho\right] \\
&\qquad \times \left[E_{\mathrm{elec}} + \varepsilon_{fs}\left(r_2 + r_1\right)^2\right] \\
&\text{The 1st level: } E_1 = \pi r_1^2 \rho E_{\mathrm{DA}} + \left[(1-\lambda)\,\pi r_1^2 \rho + (1-\lambda)^2\right. \\
&\qquad \times \pi r_2^2 \rho + \cdots \\
&\qquad \left. + (1-\lambda)^m \pi r_m^2 \rho\right] \\
&\qquad \times \left[E_{\mathrm{elec}} + \varepsilon_{\mathrm{fs}} r^2\right],
\end{aligned} \tag{10}$$

where $r_1, r_2, \ldots, r_m$ are cluster radiuses (in M different sizes), respectively, and r is used for calculating the transmission range in the 1st level, which we explain in (15). Here, E_i is the energy consumed on each cluster leader in ith level.

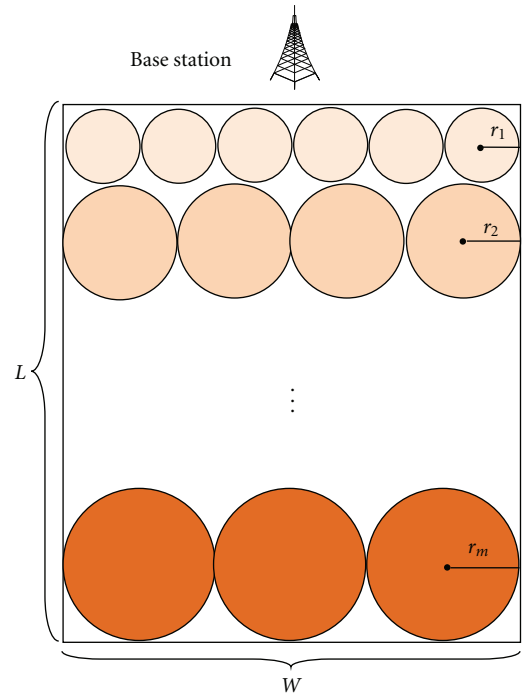


Figure 2: Calculation of cluster radius.

Because we assume that the energy consumption of cluster leaders in each level is similar, (11) is applied to calculate cluster radius in each level:

$$\begin{aligned}
&E_1 \cong E_2 \cong \cdots \cong E_m, \\
&r_1 + r_2 + \cdots r_m = \frac{L}{2},
\end{aligned} \tag{11}$$

where L is the length of sensing area (see Figure 2).

4.2. A Numerical Example. It is assumed that a BS wants to construct four clusters of different sizes, and it lets $M = 4$. The ratio of r_1, r_2, r_3, and r_4 can be obtained by (12a). Then we put the obtained ratio in (12b) to calculate the actual cluster radius:

$$\begin{aligned}
&\text{The 4th level: } E_4 = \pi r_4^2 \rho E_{\mathrm{DA}} + (1-\lambda)\,\pi r_4^2 \rho \\
&\qquad \times \left[E_{\mathrm{elec}} + \varepsilon_{\mathrm{fs}}\left(r_4 + r_3\right)^2\right] \\
&\text{The 3rd level: } E_3 = \pi r_3^2 \rho E_{\mathrm{DA}} \\
&\qquad + \left[(1-\lambda)\,\pi r_3^2 \rho + (1-\lambda)^2 \pi r_4^2 \rho\right] \\
&\qquad \times \left[E_{\mathrm{elec}} + \varepsilon_{\mathrm{fs}}\left(r_3 + r_2\right)^2\right]
\end{aligned}$$

$$\text{The 2nd level: } E_2 = \pi r_2^2 \rho E_{\text{DA}} + \left[(1-\lambda)\pi r_2^2\rho + (1-\lambda)^2\pi r_3^2\rho + (1-\lambda)^3\pi r_4^2\rho\right] \times \left[E_{\text{elec}} + \varepsilon_{\text{fs}}(r_2 + r_1)^2\right]$$

$$\text{The 1st level: } E_1 = \pi r_1^2 \rho E_{\text{DA}} + \left[(1-\lambda)\pi r_1^2\rho + (1-\lambda)^2\pi r_2^2\rho + (1-\lambda)^3\pi r_3^2\rho + (1-\lambda)^4\pi r_4^2\rho\right] \times \left[E_{\text{elec}} + \varepsilon_{\text{fs}} r^2\right], \tag{12a}$$

$$E_1 \cong E_2 \cong E_3 \cong E_4, \quad r_1 + r_2 + r_3 + r_4 = \frac{L}{2}. \tag{12b}$$

5. Intracluster Energy Balancing

In this section, we describe the strategy adopted for intracluster energy balancing in details. Firstly, two different nodes, the processor and the forwarder, will be elected as cluster leaders instead of the common CH. The election of processors considers both the residual energy and the distance between the candidates and other nodes, and their locations in each cluster are regarded as ideal. Then, clusters are formed based on the radius obtained above (see Figure 3).

5.1. Processor and Forwarder Election. At the beginning of each round of rotation, each node broadcasts message *E_Msg* (*ID*, *Energy*, $L(x, y)$) with radius r_j which includes sensor node ID, residual energy, and node coordinate (we take clusters in level j as an example ($1 \le j \le M$)). Any other node within communication radius r_j is considered as their neighbors and updates the neighbor information table after receiving messages. Every node whose residual energy is higher than E_{res}_MAX has chance to participate in the processor and forwarder competition and become a candidate. Then, each candidate will calculate the mean residual energy EM_j of all neighbors according to the updated table:

$$\text{EM}_j = \sum_{j=1}^{m} \frac{E_{\text{rj}}}{m}, \tag{13}$$

where EM_j is the mean residual energy of node S_j and m is the total number of S_j's neighbors.

It is easy to determine the mean communication distance among node S_j and its neighbor nodes:

$$d_j = \sum_{k=1, k\neq j}^{m} \frac{d_{\text{jk}}}{m}. \tag{14}$$

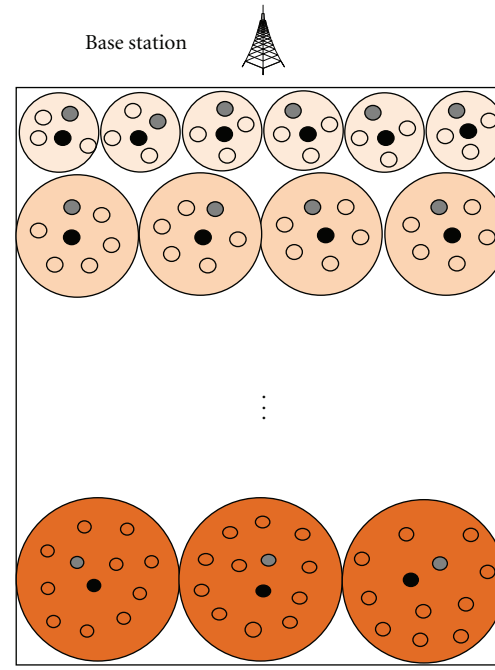


Figure 3: Election of cluster leaders.

d_{jk} is the distance between node S_j and S_k:

$$d_{\text{jk}} = \sqrt{(x_j - x_k)^2 + (y_j - y_k)^2}. \tag{15}$$

With consideration of both residual energy and distance, each candidate calculates the competition bids of being elected as processor and forwarder using (16) and (17), respectively:

$$CP_j = \chi \frac{E_{\text{rj}}}{\text{EM}_j} + \delta \frac{1}{d_j}, \tag{16}$$

$$CF_j = \chi \frac{E_{\text{rj}}}{\text{EM}_j} + \delta \frac{r_j + r_{j-1}}{d(j, BS)}. \tag{17}$$

The value of χ and δ is determined by the distribution of nodes within cluster and their residual energy situation. $d(j, BS)$ denotes the distance between S_j and the BS. And then the candidate broadcasts competition message *Com_Pro* (ID_j, Er_j, CP_j, CF_j) with radius r_j.

All the candidates are set in receive state and wait a time T. The length of T is determined to at least make sure that the nodes can receive the competition message from all its neighbors. Then each candidate compares competition bids of itself and all competition packet bids. The one with the largest value of CP_j will succeed in competition for processor while the one with the largest value of CF_j for forwarder. If the highest bids of CP_j or CF_j are even, the node with higher residual energy will be chosen. If a candidate possesses both the largest value of CP_j and CF_j, it will play the two roles at the same time.

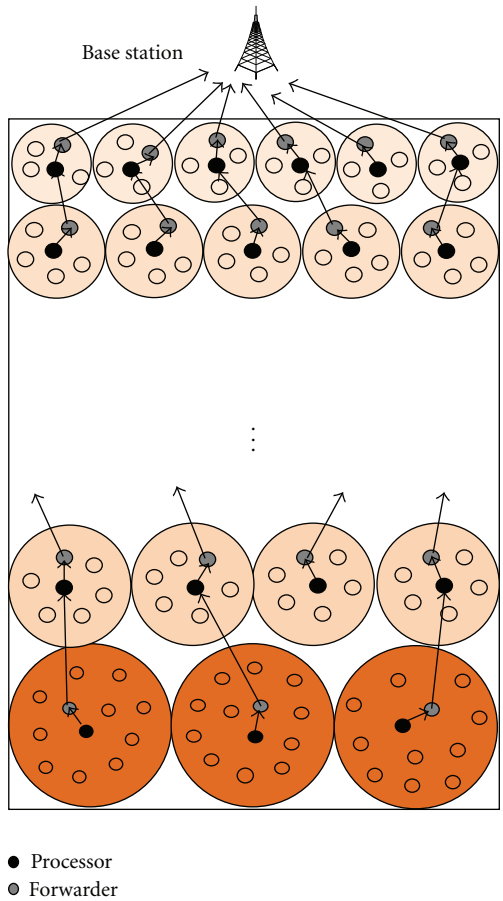


Figure 4: Transmission mode of cluster leaders.

5.2. Cluster Formation. According to the comparison results, the eligible candidate will broadcast processor competition success message *Suc_Pro* $(ID_j, Er_j, L(x_j, y_j))$ with radius r_j and, if not, wait for *Suc_Pro* $(ID_j, Er_j, L(x_j, y_j))$ message from neighbor nodes with the highest competition bids. Nodes give up competition as soon as they receive *Suc_Pro* $(ID_j, Er_j, L(x_j, y_j))$ message from neighbors. Meanwhile, they send *Join_Pro* $(ID_j, Er_j, L(x_j, y_j))$ message to neighbors with the highest transmit power. As forwarders are only responsible for forwarding the aggregated results, there is no need for broadcasting the success message of forwarder competition to all nodes in the cluster. It only adds the *Suc_For* $(ID_j, Er_j, L(x_j, y_j))$ information to the cluster-joining packet and then sends it to the processor (this step can be omitted if it is the same node). Therefore, no much overhead will be appended to the proposed algorithm compared with the existing algorithms. Then, the processor determines the TDMA slot assignment for the cluster members.

5.3. Data Aggregation Tree Construction. In this paper, the multihop algorithm considers nodes on the forwarder backbone in the forwarding direction (i.e., closer to the base station) only. Data aggregation tree generation algorithm is as follows.

We take level p and level q for example $(1 \le p < q \le M)$. Suppose the forwarder of cluster a (CF_a) in level q needs to choose a processor in level p as its relay node. Let clusters b and c in level p be the two nearest clusters from cluster a in level q. According to (1), CF_a can figure out the difference value between the energy consumption for sending its data packets to CP_b and CP_c:

$$\Delta E_T = \varepsilon_{\text{fs}} * k_a * \left(d_b^2 - d_c^2\right). \tag{18}$$

And the difference value between the residual energy of CP_b and CP_c is

$$\Delta E = E_b - E_c. \tag{19}$$

Compare ΔE with ΔE_T; if ΔE is bigger than ΔE_T, we will choose CP_b as the relay node; otherwise, we will choose CP_c. That means only if the residual energy of the candidate with longer distance to the current forwarder is much more than the closer one, the longer distance one can be chosen.

Each forwarder computes the value of ΔE and ΔE_T for the two nearest candidates respectively. It will choose as its relay (parent) node according to the calculation results. This strategy not only considers the residual energy of processors but also the total energy consumption of the whole network. Meanwhile, it considers the communication distance as well as the balance of energy consumption among all forwarders. Therefore, the overall situation of the network is taken better care of in our mechanism.

In the process of communication, each processor gathers the data from members except the forwarder within its cluster, aggregates them into one packet, and then transmits them to the forwarder. The forwarder will aggregate the compressive data with its own data, and then transmit them to its next forwarder (see Figure 4).

The proposed algorithm consists of four procedures: processor and forwarder election, cluster formation, data aggregation tree construction, and data transmission. The pseudocode of SCA is shown in Pseudocode 1.

5.4. Clusters Maintenance. As the power of cluster leaders may be exhausted quickly because of the much larger loads imposed on them, the phase of cluster maintenance is very important. In SCA, the cluster maintenance phase consists of cluster-leader rotations within a cluster and cross-level data transmission to BS.

(i) Cluster-leader rotations in a cluster: if the remaining power of any processor or forwarder is under E_{res}_MAX, a new one is elected from among other plain nodes, while a *change_msg* is broadcast to inform cluster members of the change of cluster leaders.

(ii) Cross-level data transmission to the BS: as clusters in the 1st level are the smallest in size, the process of taking turns serving as cluster leaders for nodes in it may finish quickly. Therefore, when the BS is aware that each sensor node in the 1st level can no longer serve as a cluster leader, it will broadcast a message to allow the cluster leaders in the 2nd level to transmit data to BS directly (see Figure 5). It is the same for 3rd level, 4th level,…, Mth level. In this way, the network lifetime can be prolonged.

Definitions:
CP_j: the competition bids of being elected as processor
CF_j: the competition bids of being elected as forwarder
$S_{i\text{-status}}$: the status of node S_i
$S_{i\text{-processor}}$: the status of node S_i is processor
$S_{i\text{-forwarder}}$: the status of node S_i is forwarder
$S_{i\text{-plain}}$: the status of node S_i is plain node
E: the difference value between the residual energy of two candidate forwarders
E_T: the difference value between the energy consumption for sending its data packets to two candidate forwarders

(1) **Procedure** Processor and forwarder election
(2) Each node broadcasts message packet
(3) Neighbors update the neighbor information table
(4) Each candidate calculates the competition bids CP_j and CF_j
(5) Broadcast competition message Com_Pro
(6) Receive and compare competition bids
(7) Determine processor and forwarder
(8) **end procedure**

(9) **Procedure** Cluster construction
(10) **If** $S_{i\text{-status}} \leftarrow \mathbf{S_{i\text{-processor}}}$ **then**
(11) S_i broadcasts competition success message with its radius
(12) **else if** $S_{i\text{-status}} \leftarrow \mathbf{S_{i\text{-forwarder}}}$ **then**
(13) Add the success information to the joining packet to the processor
(14) **else if** $S_{i\text{-status}} \leftarrow \mathbf{S_{i\text{-plain}}}$ **then**
(15) Send joining message to the processor
(16) **end if**
(17) **end procedure**

(18) **Procedure** Data aggregation tree construction
(19) Forwarders broadcast the cost message packet
(20) Calculate ΔE and ΔE_T
(21) Compare ΔE with ΔE_T and choose the eligible one as its relay node
(22) **end procedure**

(23) **Procedure** Data transmission
(24) **while** $S_{i\text{-status}} = \mathbf{S_{i\text{-processor}}}$ **do**
(25) **if** $\mathbf{S_{i\text{-processor}}} \leftarrow \mathbf{S_{i\text{-forwarder}}}$ **then**
(26) Aggregate all the data from members and transmit to its next forwarder
(27) **else**
(28) Receive all the data from members except the forwarder
(29) Aggregate and transmit data to the forwarder
(30) **end if**
(31) **end while**
(32) **while** $S_{i\text{-status}} = \mathbf{S_{i\text{-forwarder}}}$ **do**
(33) Receive data from the processor of its cluster
(34) Aggregate and transmit to its relay forwarder
(35) **end while**
(36) **while** $S_{i\text{-status}} = \mathbf{S_{i\text{-plain}}}$ **do**
(37) Transmit data to the processor
(38) **end while**
(39) **end procedure**

Pseudocode 1: Pseudocode of the proposed algorithm.

6. Performance Evaluation

6.1. Complexity Analysis. An analysis of the SCA algorithm is made in this section. As we can see from Figure 3, the process of processor and forwarder election is message driven; thus we first discuss its message complexity.

Lemma 1. *The message complexity of the cluster formation algorithm is $O(N)$ in the network.*

Proof. At the beginning of the processor and forwarder competition selection phase, there will be N messages $E\_Msg(ID, Energy, L(x, y))$ broadcasted by all nodes (N is the total number of sensor nodes). As each node whose residual energy is higher than E_{res}_MAX has chance to become a candidate, we assume that the ratio of eligible nodes is p. Then Np candidates are produced and each of them broadcasts a competition message $Com\_Pro(ID_j, Er_j, CP_j, CF_j)$.

Table 2: Comparison of the message complexity.

	SCA	ACT	EECA	UCR	EECS	MR-LEACH	LEACH
Message complexity	*(O)N*	*(O)N*	*(O)N*	*(O)N*	*(O)N*	*(O)N*	*(O)N*

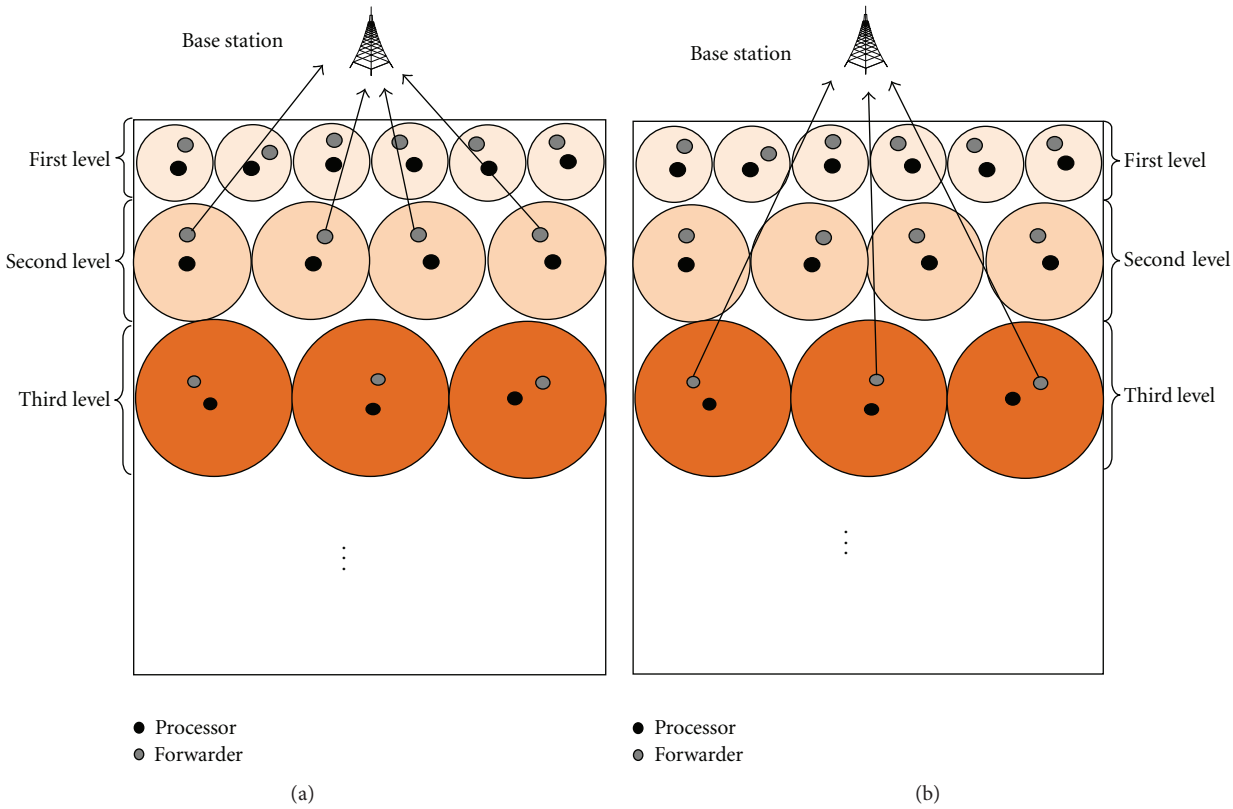


Figure 5: The architecture of cross-level data transmission.

Suppose k processors are selected, and they will send out k competition success message $Suc\_Pro(ID_j, Er_j, L(x_j, y_j))$. Accordingly, there will be N-k $Join\_Pro(ID_j, Er_j, L(x_j, y_j))$ messages sent by other nonprocessors. As forwarders only add the competition success message to the cluster-joining packet, there will not be other extra messages produced. Thus the messages add up to $N + pN + k + N - k = (p + 2)N$ at the cluster formation stage per round, that is, $O(N)$.

Table 2 provides the comparison results of the message complexity for several existing protocols. Although two different nodes, the processor and the forwarder, are elected as cluster leaders instead of the common CH, the message complexity of our proposed algorithm is not added. Clearly, our approach is better than others being used for comparison. □

6.2. Correctness Analysis

Lemma 2. *There is no chance that two nodes are both processors or forwarders if one is in the other's neighboring node set* R_{CH}.

Proof. Suppose S_u and S_v are both candidates in the cluster-leader selection phase, and S_u is in S_v's neighboring node set $S_v - R_{CH}$. According to our proposed algorithm, if S_u and S_v possess the even highest bids of CP_j or CF_j, the one which possesses higher residual energy will be chosen. The most special occasion is that the highest bids of CP_j or CF_j and the residual energy of two nodes are both the same. In this case, if S_u first becomes a leader node, then it will notice S_v its state, so S_v quits the competition and becomes an ordinary node, and vice versa. That is to say, cluster leaders are well distributed. □

6.3. Discussion

6.3.1. Percentage p. As we can see from Section 5.1, the percentage p of eligible nodes determines the number of candidates of cluster leaders. On the one hand, enough candidates guarantee good cluster-leader choosing in terms of residual energy. On the other hand, too many candidates will cause a considerable message overhead. Thus a proper value of p should be chosen in order to guarantee the quality of cluster-leader selection and reduce the message overhead.

6.3.2. Synchronization. Synchronization is another important issue needed to be paid attention to for the operation

of SCA. It is assumed that all sensor nodes are synchronized and start the clustering phase at the same time. We can achieve it, for instance, by having the base station periodically broadcast synchronization pulses. Readers can obtain more details about the time synchronization issue in clustered wireless sensor networks with reference to [22].

6.3.3. Delay and throughput. The election of two nodes as cluster leaders will have some impact on delay and throughput of the whole network. Processors will transmit the processed data to their forwarders after their collection and aggregation instead of transmitting them directly to the next relay forwarder. This forwarding process takes a not long but certain time, so it would imply waiting longer at next-aggregation points and delaying the final delivery. Accordingly, the throughput will decrease. However, the adopted cross-level data transmission mode in the later phase will reduce the latency as well as increase the throughput, which will compensate for the total performance degradation.

7. Analysis of Energy Consumption

As outlined in Sections 4 and 5, the total energy consumed per round can be divided into two distinct phases which consist of the cluster set-up phase and the data transfer phase. The mathematical expressions that calculate an estimation of the energy consumed in each phase are provided, which we use to evaluate whether the loads are more balanced by adopting task separation. According to (1)–(3), we can obtain (20)–(27) as follows.

7.1. Clustering Phase. As described in Section 5, each round consists of creating a dominating set of cluster leaders chosen from a certain amount of candidates. We assume that the ratio of eligible nodes is p, then Np candidates are produced and each of them will broadcast a competition message. We take a cluster in level j as an example ($1 < j < M$). Assuming that the length of one message is l bytes and $A = L \times W$, then the energy consumed by candidates in a cluster per round is given by

$$E_c^1 = Npl \times \frac{\pi r_j^2}{A}\left(E_{\text{elec}} + \varepsilon_{\text{fs}} r_j^2\right) + ql \times \frac{\pi r_j^2}{A} \times E_{\text{elec}}, \quad (20)$$

where the first term represents the energy consumed for transmitting competition messages sent by cluster-leader candidates. The second term signifies the energy consumed in receiving the compete messages from other cluster-leader candidates within the competition radius. The number of messages received is based on the estimate that q cluster-leader candidates will fall within the competition radius.

Suppose that k processors are selected and each of them will send out a competition success message within radius r_j. Thus the energy consumed in each cluster for the processor advertisement message will be

$$E_c^2 = k \times \frac{\pi r_j^2}{A} \times l \times \left(E_{\text{elec}} + \varepsilon_{\text{fs}} r_j^2\right). \quad (21)$$

As there are $N-k$ nonprocessor nodes, each of which will receive this message and then sends a Join_Pro message to its processor; energy consumed during this process will be

$$E_c^3 = (N-k) \times \frac{\pi r_j^2}{A} \times l \times E_{\text{elec}} + (N-k) \times \frac{\pi r_j^2}{A} \times l \times \left(E_{\text{elec}} + \varepsilon_{\text{fs}} r_j^2\right). \quad (22)$$

Finally, each processor will receive these Join_Pro messages and the amount of energy consumed will be

$$E_c^4 = (N-k) \times \frac{\pi r_j^2}{A} \times l \times E_{\text{elec}}. \quad (23)$$

As forwarders only add the competition success message to the cluster-joining packet, there will not be other extra messages produced and then more energy consumed.

7.2. Data Transmission Phase. In the data transmission phase, each plain node sends a single data message of t bytes to the processor, and the energy consumed is

$$E_c^5 = (N-k) \times \frac{\pi r_j^2}{A} \times t \times \left(E_{\text{elec}} + \varepsilon_{\text{fs}} r_j^2\right). \quad (24)$$

Then each processor will receive these data messages:

$$E_c^6 = (N-k) \times \frac{\pi r_j^2}{A} \times t \times E_{\text{elec}}. \quad (25)$$

Next, each processor will aggregate the messages of its own cluster and relayed from its above level:

$$E_c^7 = \left[(N-k) \times \frac{\pi r_j^2}{A} \times t + D_{\text{relay}}^j\right] E_{\text{DA}}, \quad (26)$$

where D_{relay}^j represents the amount of data relayed from level $j+1$ to level j. Since processors and forwarders in the same cluster are very close to each other, energy consumed can be considered negligible in the local forwarding process. Finally, forwarders will transmit these data to their next relay nodes:

$$E_c^8 = (1-\lambda)\left[(N-k) \times \frac{\pi r_j^2}{A} \times t + D_{\text{relay}}^j\right] \times \left[E_{\text{elec}} + \varepsilon_{\text{fs}}\left(r_j + r_{j-1}\right)^2\right]. \quad (27)$$

From the equations given above, we can summarize the total energy consumed in each round by each processor and

forwarder in level j, respectively: $E_{\text{pro}} = E_c^{2'} + E_c^{4'} + E_c^{6'} + E_c^{7'}$, $E_{\text{for}} = E_c^{3'} + E_c^{8'}$,

$$\begin{aligned} E_{\text{pro}} &= l \times \left(E_{\text{elec}} + \varepsilon_{\text{fs}} r_j^2\right) + (N-k) \times \frac{\pi r_j^2}{A} \times l \times E_{\text{elec}} \\ &\quad + (N-k) \times \frac{\pi r_j^2}{A} \times t \times E_{\text{elec}} \\ &\quad + \left[(N-k) \times \frac{\pi r_j^2}{A} \times t + D_{\text{relay}}^j\right] E_{DA}, \\ E_{\text{for}} &= l \times E_{\text{elec}} + l \times \left(E_{\text{elec}} + \varepsilon_{\text{fs}} r_j^2\right) + (1-\lambda) \\ &\quad \times \left[(N-k) \times \frac{\pi r_j^2}{A} \times t + D_{\text{relay}}^j\right] \\ &\quad \times \left[E_{\text{elec}} + \varepsilon_{\text{fs}} \left(r_j + r_{j-1}\right)^2\right]. \end{aligned} \tag{28}$$

Let $Y = (N-k) \times (\pi r_j^2 / A)$, then

$$\begin{aligned} E_{\text{pro}} &= l \times \left(E_{\text{elec}} + \varepsilon_{\text{fs}} r_j^2\right) + Y \times l \times E_{\text{elec}} + Y \times t \\ &\quad \times E_{\text{elec}} + \left(Y \times t + D_{\text{relay}}^j\right) E_{\text{DA}}, \\ E_{\text{for}} &= l \times \left(E_{\text{elec}} + \varepsilon_{\text{fs}} r_j^2\right) + l \times E_{\text{elec}} + (1-\lambda) \\ &\quad \times \left[Y \times t + D_{\text{relay}}^j\right] \left[E_{\text{elec}} + \varepsilon_{\text{fs}} \left(r_j + r_{j-1}\right)^2\right]. \end{aligned} \tag{29}$$

In order to estimate the energy consumption of each processor and forwarder in one round, we consider the difference of their consumed energy. Our original goal is to lighten the load of CHs by task separation. If energy consumption of a couple of processor and forwarder in the same cluster in each round is nearly equal, the energy consumption of CHs is slowed down to half its common values, thus prolonging network lifetime to the maximum extent. Then, we have

$$\begin{aligned} E_{\text{for}} - E_{\text{pro}} &= \left(Yt + D_{\text{relay}}^j\right) \\ &\quad \times \left\{(1-\lambda) \left[E_{\text{elec}} + \varepsilon_{\text{fs}} \left(r_j + r_{j-1}\right)^2\right] - E_{\text{DA}}\right\} \\ &\quad - (Y-1)\, l E_{\text{elec}} - Y t E_{\text{elec}}. \end{aligned} \tag{30}$$

As it is an equation whose highest order is quartic (r_j^4), it is not easy to observe their difference intuitively. So we randomly take a distribution whose total level $M = 5$ as an example. Assuming $j = 3$, we calculate the value of $E_{\text{for}} - E_{\text{pro}}$.

Table 3: Calculation parameters.

Parameter	Value	Unit
N	98	
A	9800	m^2
K	20	
λ	0.15	
r_2	7.12	m
r_3	11.36	m
r_4	18.12	m
r_5	28.91	m
E_{elec}	50	nJ/bit
ε_{fs}	10	pJ/bit/m^2
E_{DA}	5	nJ/bit/signal
$l(t)$	1	byte

Combining with Tables 5 and 6, the parameters used are listed in Table 3:

$$\begin{aligned} Y &= (N-k) \times \frac{\pi r_j^2}{A} = 3.19, \\ D_{\text{relay}}^j &= D_{\text{relay}}^3 = (1-\lambda)^2 \pi r_5^2 \rho + (1-\lambda)\, \pi r_4^2 \rho = 27.74, \\ E_{\text{for}} - E_{\text{pro}} &= (8Y + 27.74) \left\{0.85 \left[50 + 0.01 \times (7.12 + 11.36)^2\right] - 5\right\} \\ &\quad - 8\,(Y-1) \times 50 - 8Y \times 50 \\ &= (8Y + 27.74) \times 40.4 - 800Y + 400 \\ &= -0.3\ (nJ). \end{aligned} \tag{31}$$

From the above calculations we can see that, for a network which has 5 levels, the energy consumption difference between a couple of processor and forwarder in the middle level is only 0.3 nJ. That is to say, the consumed energy of each processor and its corresponding forwarder is nearly equal, and the processor really works for spreading the load. So we obtain the expected results.

8. Simulations

We conduct simulations to study the performance of our proposed energy balancing algorithm. First of all, we describe the simulation settings. Secondly, simulation results are presented showing the performance results under different performance metrics. Finally, we discuss and analyze the simulation results. Table 4 provides a comparison of the related work with respect to different clustering attributes, from which we choose LEACH, EECA, MR-LEACH, and ACT for comparison.

8.1. Simulation Environment. We analyze the performance of SCA algorithm by Omnet++ which allows efficient and realistic modeling of sensor nodes by using an integrated technical

Table 4: Comparison of related works with respect to clustering attributes.

Clustering protocol	Cluster-leader selection	Cluster formation	Cluster size	Data aggregation tree construction	Energy dissipation
LEACH	Random	Closest CH	Equal	No	Unbalanced
MR-LEACH	Nodes with the largest residual energy	Closest CH	Equal	BS helps to choose relay CHs for lower layer CHs	Somewhat balanced
EECS	Random with election	Closest based using three parameters	Unequal	No	Somewhat balanced
UCR	Random with election	Closest CH	Unequal	Greedy geographic forwarding algorithm based on relay path cost	Somewhat balanced
EECA	Based on two parameters	Closest CH	Equal	Based on a weight function	Relatively balanced
ACT	Ideal location	Closest CH	Unequal	Equal allocation of the relay loads	Relatively Balanced
SCA	Based on two parameters	Closest CH	Unequal	Based on the comparison of transmission and residual energy	Balanced

Table 5: Simulation parameters.

Parameter	Value
Number of nodes	98
Network scale (m^2)	70×140
Location of BS (m)	(40, 150)
Initial energy of each node (J)	2
The ratio of candidates p	0.3
E_{elec} (nJ/bit)	50
ε_{fs} (pJ/bit/m^2)	10
d_0 (m)	100
E_{DA} (nJ/bit/signal)	5
Data packet size (bit)	500

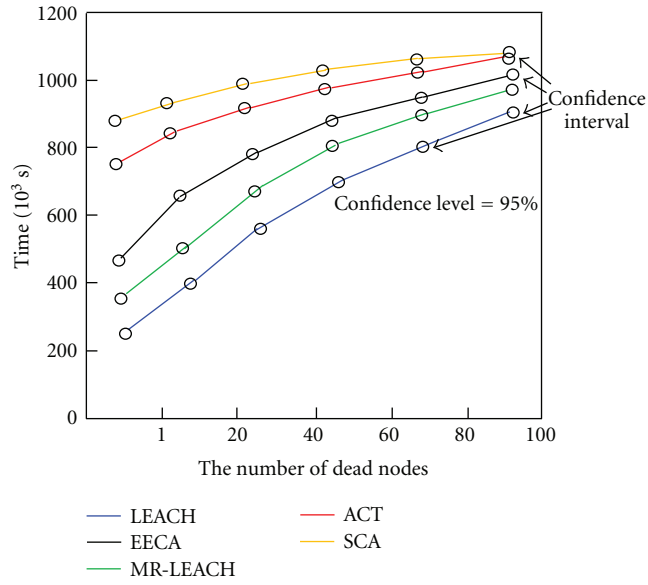


Figure 6: Network lifetime.

computing environment. Because this paper focuses on energy-efficient and balanced routing in the network layer, an ideal MAC layer and error-free communication links are assumed for simplicity. We perform the simulation study under steady state, and the other parameters of the simulation are listed in Table 5.

8.2. Results and Analysis

8.2.1. Network Lifetime. First of all, we measure the lifetime of network. Figure 6 gives the number of living nodes over time. As evident from the figure, SCA has a longer network lifetime than LEACH, EECA, MR-LEACH, and ACT. As for LEACH, each sensor node elects itself as a CH with some probability with no regard to the residual energy. Moreover, all CHs communicate with the BS directly in LEACH which leads to high energy consumption in communication and thus shorting the network lifetime. Even though the CHs in EECA and MR-LEACH are selected from sensor nodes with sufficient power, their use of multihop communications increases the burden of cluster heads near the BS. As the CHs close to the BS share higher relaying loads, their energy would be used up faster and die earlier. ACT considers the adjustment of cluster sizes during data relay, but the same disadvantages with EECA and MR-LEACH still exist in it, so that it performs better than the two but worse than SCA. When the data is relayed among clusters, the cluster sizes are adjusted and the task of a CH is allocated to two nodes in SCA, which reduces the energy consumption of critical nodes; as a result, SCA achieves the longest network lifetime.

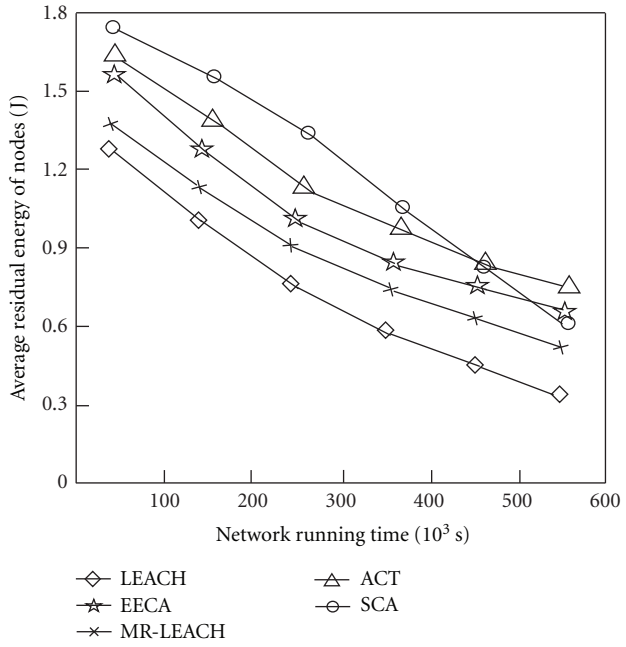


Figure 7: Average residual energy.

8.2.2. Average Residual Energy. Figure 7 compares the average residual energy of nodes of the five algorithms. We can observe that average residual energy of nodes under SCA algorithm is greater than that of the other four algorithms. LEACH adopts single-hop communications with the CH sending its data directly to the BS leading to its lower average residual energy. EECA, MR-LEACH, ACT, and SCA utilize multihop communications that require less energy consumption from each sensor node. With consideration of cluster size and cluster-leader production, SCA balances the load on each cluster and alleviates the burden of those critical nodes. In addition, $(\Delta E - \Delta E_T)$ is defined as the metric to choose relay nodes during data aggregation tree construction, which considers the total energy consumption of the whole network. In this way, energy spent by sensor nodes close to the BS is less than in EECA, MR-LEACH, and ACT, so the average energy dissipation in SCA is lower than that in the other four. But as time goes on, more and more clusters need to transmit their data to the BS directly in SCA. Forwarders bear most of the tasks at the later phase, and the effect of processors is not outstanding now. However, selecting two nodes for cluster leaders will consume more extra energy. Therefore, it increases the average energy dissipation in SCA and leads to more energy consumption than ACT and EECA after running for approximately $460/10^3$ s.

8.2.3. The Standard Deviation of Energy Consumption of Cluster Leaders. Figure 8 compares the standard deviation of energy consumption of cluster leaders in LEACH, EECA, MR-LEACH, ACT, and SCA. The CHs in LEACH are picked out randomly, providing each sensor node a chance to serve as a CH. Accordingly, the standard deviations of energy consumption of CHs in LEACH show substantial variations. MR-LEACH considers only the residual energy of nodes,

Table 6: Variations of cluster radius with level M.

Level M	r_1	r_2	r_3	r_4	r_5
$M = 3$	13.81	21.72	34.46	×	×
$M = 4$	7.69	12.16	19.34	30.81	×
$M = 5$	4.49	7.12	11.36	18.12	28.91

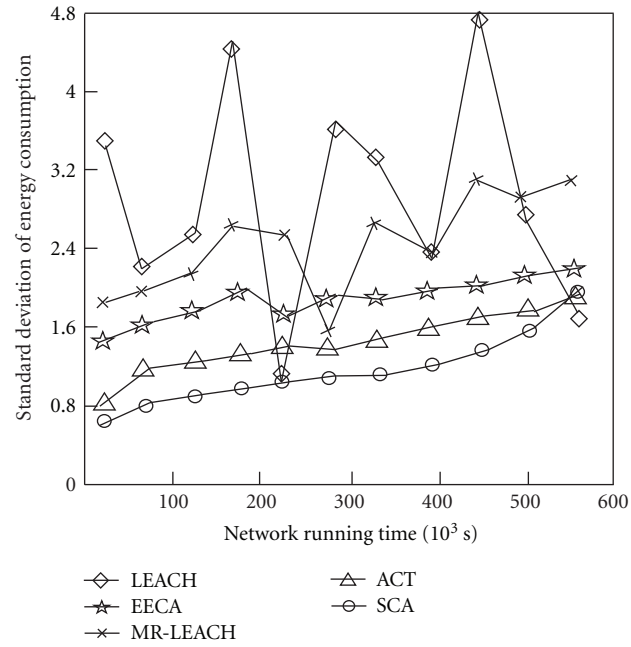


Figure 8: The standard deviation of energy consumption of cluster leaders.

and EECA chooses the CHs based on residual energy and distance. Thus their curves display irregular oscillation in each round. The CHs in ACT are chosen according to the ideal locations calculated depending on the load balance, and its curve of standard deviation for energy consumption is relatively steady. SCA calculates cluster sizes according to the loads on CHs to balance the given loads on each CH. Meanwhile, it separates tasks of one single cluster head to two nodes, which further balances the energy consumption of cluster leaders. As a result, the value of standard deviation of energy consumption in SCA is minimized.

Table 6 shows the variation of cluster radius with level M within the same scenario, which is consistent with our design idea in Section 4.1.

8.2.4. Variance of Average Residual Energy. Figure 9 shows the experiment result of variance of average residual energy of nodes in network and reflects the proportionality of network energy consumption. As to LEACH algorithm, most energy loads concentrate on CHs and the excessive energy consumption leads to early death of them, thus causing much more uneven distribution of node energy in network than all others being compared. In EECA, MR-LEACH, and ACT, the shorter the distance between cluster leaders and the BS, is the much heavier burden the leaders will have. In addition, even though ACT considers arranging cluster sizes based on

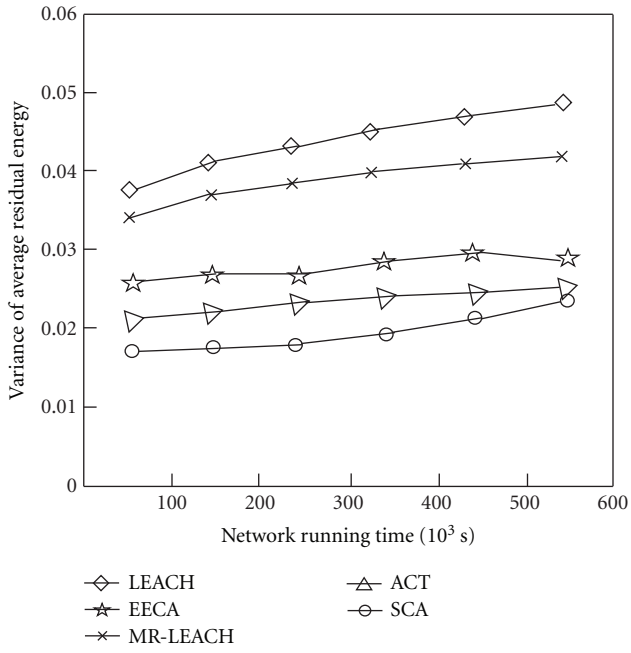


Figure 9: Variance of average residual energy.

the energy consumption, the selection of a new CH makes the locations of CHs deviate from the original ideal ones. All these make the scatterings of energy consumption oscillating. In comparison, SCA has a better and more stable value in the early phase, but in the later phase cross-level data transmission directly to the BS and the election of two new cluster leaders leads to the rapid reduction of energy.

9. Conclusions

In this paper, we focus on the problem of unbalanced energy dissipation when employing the multihop routing in a cluster-based WSN. We propose an approach that balances the energy consumption among clusters and slows the energy consumption of CHs. For intercluster energy balancing, a cluster radius is calculated with consideration of the relaying load undertaken by each cluster, thus balancing the energy dissipation among clusters. As for intracluster energy dissipation, a separating mode is adopted to alleviate the burden of critical nodes in each cluster and prolong the network time. Simulation results show that our method outperforms LEACH, MR-LEACH, EECA, and ACT in aspects of network lifetime, energy efficiency, and balanced extent of energy dissipation.

Acknowledgments

The authors would like to thank the anonymous reviewers and the editor for their constructive comments on this article. This work is supported by Program for Changjiang Scholars and Innovative Research Team in University (no. IRT1078), the Key Program of NSFC-Guangdong Union Foundation (U1135002), Major national S&T program (2011ZX03005-002), the National Natural Science Foundation of China (no. 61272541, 61202389), and the Fundamental Research Funds for the Central Universities (no. JY10000903001).

References

[1] P. Medagliani, J. Leguay, G. Ferrari, V. Gay, and M. Lopez-Ramos, "Energy-efficient mobile target detection in wireless sensor networks with random node deployment and partial coverage," *Pervasive and Mobile Computing*, vol. 8, no. 3, pp. 429–447, 2012.

[2] N. Aslam, W. Phillips, W. Robertson, and S. Sivakumar, "A multi-criterion optimization technique for energy efficient cluster formation in wireless sensor networks," *Information Fusion*, vol. 12, no. 3, pp. 202–212, 2011.

[3] A. F. Liu, P. H. Zhang, and Z. G. Chen, "Theoretical analysis of the lifetime and energy hole in cluster based wireless sensor networks," *Journal of Parallel and Distributed Computing*, vol. 71, no. 10, pp. 1327–1355, 2011.

[4] R. Zhu, Y. Qin, and J. Wang, "energy-aware distributed intelligent data gathering algorithm in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2011, Article ID 235724, 13 pages, 2011.

[5] K. Iwanicki and M. Van Steen, "Gossip-based self-management of a recursive area hierarchy for large wireless sensornets," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 4, pp. 562–576, 2010.

[6] T. Kim, Y. Lee, J. Sung, and D. Kim, "Hierarchical network protocol for large scale wireless sensor networks," in *Proceedings of the 7th IEEE Consumer Communications and Networking Conference (CCNC '10)*, pp. 1–2, Las Vegas, Nev, USA, January 2010.

[7] D. Kumar and R. B. Patel, "Multi-hop data communication algorithm for clustered wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2011, Article ID 984795, 10 pages, 2011.

[8] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.

[9] X. Fan and Y. Song, "Improvement on LEACH protocol of wireless sensor network," in *Proceedings of the International Conference on Sensor Technologies and Applications, (SENSORCOMM '07)*, pp. 260–264, Valencia, Spain, October 2007.

[10] V. Loscri, G. Morabito, and S. Marano, "A two-levels hierarchy for low-energy adaptive clustering hierarchy (TL-LEACH)," in *Proceedings of the 62nd IEEE Vehicular Technology Conference (VTC '05)*, pp. 1809–1813, September 2005.

[11] M. B. Yassein, A. Al-zou'bi, Y. Khamayseh, and W. Mardini, "Improvement on leach protocol of wireless sensor network (vleach)," *International Journal of Digital Content Technology and Its Applications*, vol. 3, no. 2, pp. 132–136, 2009.

[12] M. O. Farooq, A. B. Dogar, and G. A. Shah, "MR-LEACH: multi-hop routing with low energy adaptive clustering hierarchy," in *Proceedings of the 4th International Conference on Sensor Technologies and Applications, (SENSORCOMM '10)*, pp. 262–268, Venice, Italy, July 2010.

[13] M. Ye, C. F. Li, G. H. Chen, and J. Wu, "EECS: an energy efficient clustering scheme in wireless sensor networks," in *Proceedings of the 24th IEEE International Performance Computing and Communications Conference (IPCCC '05)*, pp. 535–540, 2005.

[14] G. Chen, C. F. Li, M. Ye, and J. Wu, "An unqual cluster based routing protocol in wireless sensor networks," *Wireless Network*, vol. 15, no. 2, pp. 193–207, 2009.

[15] F. Bouabdallah, N. Bouabdallah, and R. Boutaba, "On balancing energy consumption in wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 6, pp. 2909–2924, 2009.

[16] J. S. Li, H. C. Kao, and J. D. Ke, "Voronoi-based relay placement scheme for wireless sensor networks," *IET Communications*, vol. 3, no. 4, pp. 530–538, 2009.

[17] W. K. Lai, C. F. Fan, and L. Y. Lin, "Arranging cluster sizes and transmission ranges for wireless sensor networks," *Information Sciences*, vol. 183, no. 1, pp. 117–131, 2012.

[18] U. J. Jang, S. G. Lee, and S. J. Yoo, "Optimal wake-up scheduling of data gathering trees for wireless sensor networks," *Journal of Parallel and Distributed Computing*, vol. 72, no. 4, pp. 536–546, 2012.

[19] C. Sha, R. C. Wang, H. P. Huang, and L. J. Sun, "Energy efficient clustering algorithm for data aggregation in wireless sensor networks," *Journal of China Universities of Posts and Telecommunications*, vol. 17, no. 2, pp. 104–122, 2010.

[20] Z. Liu, Q. Zheng, L. Xue, and X. Guan, "A distributed energy-efficient clustering algorithm with improved coverage in wireless sensor networks," *Future Generation Computer Systems*, 2011.

[21] E. Zeydan, D. Kivanc, C. Comaniciu, and U. Tureli, "Energy-efficient routing for correlated data in wireless sensor networks," *Ad Hoc Networks*, vol. 10, no. 6, pp. 962–975, 2012.

[22] S. el Khediri, A. Kachouri, and N. Nasri, "Diverse synchronization issues in wireless sensor networks," in *Proceedings of the International Conference on Microelectronics (ICM'11)*, pp. 1–6, Hammamet, Tunisia, 2011.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 372982, 8 pages
http://dx.doi.org/10.1155/2013/372982

Research Article

Power Load Distribution for Wireless Sensor and Actuator Networks in Smart Grid Buildings

Junghoon Lee and Gyung-Leen Park

Department of Computer Science and Statistics, Jeju National University, Jeju-Do 690-756, Republic of Korea

Correspondence should be addressed to Gyung-Leen Park; glaprk@jejunu.ac.kr

Received 22 October 2012; Accepted 24 December 2012

Academic Editor: Jiman Hong

This paper presents a design and analyzes the performance of an actuator operation scheduler for wireless sensor and actuator networks, aiming at efficiently managing power consumption and distributing peak load in smart grid buildings. To create a schedule within an acceptable response time, a genetic algorithm is designed, and the scheduler places the operations of activated tasks to appropriate time slots in the allocation table. For genetic operations, each schedule is encoded to an integer-valued vector, where each element represents either start time or binary allocation map of the associated task according to the task type. The fitness function evaluates the schedule quality by estimating the load of the peaking slot. Out-task model defines *P-Penalty* and *N-Penalty* to account for the extrapower load brought by the delayed start of task operation. The performance measurement results obtained from a prototype implementation reveal that our genetic scheduler reduces the peak load by up to 35.2% for the given parameter set compared with the *Earliest* scheduling scheme, intelligently compromising two conflicting requirements of even load distribution and small initiation delay.

1. Introduction

It is too much well known that wireless sensor networks, or WSNs in short, are extending their application areas even to harsh environments such as battlefields and chemical process facilities. Moreover, they are commonly integrating a variety of actuators specific to the system goal, forming wireless sensor and actuator networks (WSANs). Their typical application is real-time monitor-and-control of critical systems. In the mean time, the smart grid is apparently also a promising target of WSANs. As a future power system, the smart grid pursues continuous monitoring, pervasive communication, self-healing reliability, and timely reaction to meet the growing demand for sustainable and clean electric energy [1]. In the smart grid, WSANs provide essential infrastructure for automatic metering and remote system monitoring. There are many standard WSAN technologies available to the smart grid including Bluetooth, Wi-Fi, ultrawideband, and Zigbee [2]. Upon those protocols, many different energy-related applications can work without human intervention, further benefiting from M2M technologies [3].

In WSANs, sensor applications capture the current status of the monitoring target via sensor readings, while the actuators are triggered by the control logic. Every sensor and actuator operation accompanies power consumption. Power management has long been a major issue in WSANs and achieved great improvement especially in wireless communication [4]. For example, Zigbee devices are known to be able to last for several years without battery replacement [5]. Hence, the requirement on power efficiency is gradually moving from sensor part to actuator part. Particularly, in smart grid buildings, heaters or air conditioners are turned on or off based on continuous monitoring of environment variables such as temperature [6]. The simultaneous operation of multiple actuators may lead to sharp increase in power consumption, deteriorating the frequency quality and thus jeopardizing the safe operation of the power supply system [7]. The scheduling-based coordination of actuator operations can shift the power load to light-loaded time interval, alleviating the above-mentioned problems.

In the smart grid perspective, the energy-efficient actuator schedule belongs to the demand response. Not only

actuator operations but also the execution of sensing applications themselves require power consumption. In smart buildings which are equipped with many actuator devices such as lighting facilities, elevator management, HVAC (Heating, Ventilation, and Air Conditioning) systems, and electric pumps, peak load shifting is more important [6]. Furthermore, charging facilities for electric vehicles will be installed in buildings. For actuator operation scheduling, time window is divided into fixed-length time slots, and each operation is aligned with the slot. Namely, actuators can be started, suspended, or resumed at each slot boundary. In addition, the load profile or interchangeably consumption profile contains the sequence of power demand along the time axis for each actuator operation. Currently, information on power consumption statistics is available in public for diverse appliances [8].

The scheduling problem for actuator operations is quite similar to real-time task scheduling in that tasks have their own deadlines [9]. Each actuator operation can be taken as a processing task while tasks arrive when the control logic decides the control action. At each arrival, task scheduling is performed. However, in actuator scheduling, tasks can run in parallel within the provisioned power cable capacity. Here, by task scheduling, some actuator operations can be delayed while others start at the moment they are activated by the control logic. The delayed start may lead to the change in power consumption dynamics. For example, the delayed initiation of air conditioners or heaters can possibly extend the operation length or increase the amount of power consumption. The scheduler must take into account this factor in schedule generation. However, such task scheduling is in most cases a complex time-consuming problem quite sensitive to the number of tasks, so conventional optimization schemes are impractical due to their extremely long execution time.

For better responsiveness, suboptimal search techniques are indispensable in spite of optimality loss. Genetic algorithms are one of the most widely used suboptimal search techniques in many different areas, not restricted to just engineering problems [10]. It can generate a task schedule within an acceptable time bound, and its execution time is even controllable by adjusting the number of genetic iterations. Moreover, it can combine a variety of efficient heuristics such as initial population selection. In this regard, this paper designs an actuator operation scheduling scheme for WSANs based on genetic algorithms. To this end, first, it is necessary to encode a schedule to a chromosome, which is represented by an integer-valued vector. Second, a fitness function must be defined to evaluate the quality of each schedule, accounting for different task behaviors according to the delayed start time. Finally, genetic operators, such as selection, crossover, and mutation, are tailored for the schedule generation based on the given task model.

This paper is organized as follows: after issuing the problem in Section 1, Section 2 surveys the background and related work of this paper. Section 3 explains actuator operation scheduler, focusing on encoding scheme design, the fitness function definition, and genetic operation customization. After performance measurement, results are demonstrated and discussed in Section 4; Section 5 finally summarizes and concludes the paper with a brief introduction of future work.

2. Background and Related Work

Reference [1] overviews the promising applications of WSNs for electric power systems, where timely information is essential for reliable power transmission and distribution from generation units to end users. Those applications include wireless automatic meter reading, remote system monitoring, equipment fault diagnostics, and the like. WSNs in the smart grid are expected to efficiently cope with harsh environmental conditions, reliability and latency requirement, packet errors with variable link capacity, and resource constraint. It focuses on the measurement of the WSN link quality in the electric-power-system environment in terms of background noise, channel characteristics, and 2.4 GHz band attention. Hence, extensive field tests have been conducted on IEEE 802.15.4-compliant sensor nodes. The experiment environments include a 500 kV substation, a main power control room, and an underground network transformer vault. After all, the wireless channel has been modeled using a log-normal shadowing path-loss model.

Reference [11] considers joint problems of control and communication in WSANs for building control systems, where WSAN serves as components of control loops. It focuses on controlling the environment variables, such as temperature, humidity, and illumination by means of heating, air conditioning, ventilating, and lighting. This system is built on top of 3-tier design consisting of network, control, and user interface. The authors propose both centralized and distributed control schemes, respectively. For both of them, Kalman filters compensate for packet losses and delays in wireless channels to estimate the current WSAN status, while the control action is decided by means of the control objective function. The performance comparison between centralized and distributed schemes is conducted in terms of control performance, energy efficiency, computational complexity, and packet loss rate. However, their control decision procedure does not consider explicitly the power consumption dynamics in actuators.

Reference [5] addresses that WSNs will play a key role in the deployment of the smart grid towards residential premises, hosting various demand and energy management applications. The authors evaluate the performance of in-home energy management schemes focusing on the energy cost reduction. This scheme allows its applications to be flexibly built on top of wireless sensor networks in home area. Here, the Zigbee technology is exploited for communication between the energy management unit and other power entities such as appliances, smart meters, and storage devices. The management system also incorporates SEP (Smart Energy Profile) 2.0 developed by Zigbee Alliance for the standardized message format and exchange procedure in automatic metering applications. The performance of the sensor network part depends on the packet size generated by the monitoring application. The simulation-based experiment finds out that

the smaller packet size is preferred for smart grid applications from the viewpoint of packet delivery ratio, delay, and jitter. Anyway, standard WSNs can be seamlessly integrated into smart grid systems.

Reference [12] presents a strategy for deep demand response of power loads considering the availability of renewable energies. Its monitoring and controlling architecture also employs WSNs, where each sensor is attached to a low-power wireless mote running an IPv6-compatible networking layer. A set of motes forms an ad hoc network while a laptop computer provides a gateway to the global network. The authors introduce the notion of slack, which means *the potential of an energy load to be advanced or deferred without affecting earlier or later operations*. According to the operation model integrating the slack, the distribution of appliance operations can lead to a better match of energy consumption to generation. The thermostatically controlled load is the main target of the time-dependent power consumption management, and this scheme mainly focuses on the coordination of renewable energy. Their power consumption model is reasonable, and our scheme will adapt it for explicit appliance scheduling.

In the mean time, our research team has been conducting researches on demand response schemes for the purpose of applying the research results promptly in Jeju area, which has established one of the world's largest smart grid testbeds [13]. Main focus is put on efficient scheduling of power device operations and charging of electric vehicles in smart homes, buildings, and charging stations. Based on the task model consisting of preemptive and nonpreemptive tasks, our strategies build an operation schedule for the given time window either by genetic algorithms or by exhaustive searches combined with a heuristic [14]. They are mainly concerning how to reduce peak load by distributing each task operation as evenly as possible. In addition, for the efficient integration of renewable energies in a smart grid unit, a dual battery management scheme decides when to charge or discharge each battery according to the availability of wind power generation and power demand approximation [15]. Until now, the power consumption is assumed to be constant irrespective of its start time.

3. Actuator Operations Scheduler Design

3.1. System and Task Models. Figure 1 depicts our system model which is inherited from the typical wireless WSAN architecture. As our design is targeted at smart grid buildings, sensors and actuators are selected for building environment control. After the sensor data analysis, a series of control actions are determined. The actuator control process notifies the power scheduler of those tasks. Each action has its own time constraint that must be completed within a specific time instant. Its power consumption is specified by load profile which is a sequence of power amount consumed on each time interval. At each task arrival, the power scheduler generates a new schedule or modifies an existing schedule considering the load profile and already admitted tasks. According to the schedule, the switch controller unit turns on or off power switches which connect respective electric devices to the power supply, be it the main power line or local renewable energies.

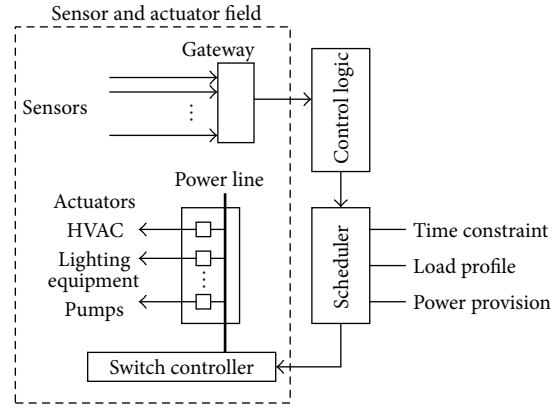


Figure 1: System model.

The operation schedule is denoted by a $N \times M$ time table, where N is the number of tasks, and M is the number of time slots. For task specification, task T_i can be modeled by the tuple of $\langle F_i, A_i, D_i, U_i \rangle$. First, F_i indicates whether T_i is preemptive or nonpreemptive. In addition, A_i is the activation time of T_i, D_i is the deadline, and U_i denotes the operation length, which corresponds to the length of the consumption profile entry. A nonpreemptive task can start from its activation to the latest start time, which can be calculated by subtracting U_i from D_i. Actually, $D_i - U_i$ is equivalent to the slack of T_i. When a start time is selected, the profile entry is just copied to the allocation table one by one, as the task must not be preempted once it has started. In contrast, the preemptive task case is quite complex. To meet its time constraint, U_i out of $(D_i - A_i)$ slots are picked for device operation. The operation can be intermittent.

The load power profile is practical for characterizing the power consumption behavior of each task. As an example, power consumption for ventilation process depends on the selected program. Along the time axis, the activated set of devices will be different stage by stage, and accordingly the amount of power consumption is different for each stage. The scheduler can assume that the power requirement of each operation step is known in priori. Here, the power consumption pattern for every electric device is aligned to the fixed-size time slot. Actually, each actuator has its own time scale in its power consumption. However, we can take the average value during each time slot considering voltage regulating equipments commonly available in most buildings. The length of a time slot can be tuned according to the system requirement on the schedule granularity and the computing time. This length usually coincides with the period of price signal change in power trade systems.

In addition to the standard consumption behavior specified by the load profile, we define two parameters to adapt the consumption dynamics according to the delayed start of actuator operations. First, *P-Penalty* represents the increase

in the power consumption for each slot of delay. If *P-Penalty* is 5% and 2 slots are delayed, the power consumption increases by 10% in every slot. Second, *N-Penalty* denotes the amount of delay which leads to one slot extension in operation length. If *N-Penalty* is 3 and the actuator operation is delayed by 6 slots, the operation length increases by 2 slots. The power consumption in the extended slots is equal to the per-slot average. Additionally, if it is delayed by 4 slots and the average per-slot power consumption is 3, the operation length increases by 2 slots. Here, the power consumption in the first-extended slot is 3, and that in the second slot is 1, which corresponds to a third of the average per-slot consumption. This adaptation model accounts for the common behavioral tendency that the power consumption largely increases in proportion to the amount of delay [12].

3.2. Encoding and Fitness Function. In the development of genetic scheduler, it is necessary to represent an allocation table by a chromosome or an integer-valued vector. Our design takes N integers for a schedule, where each element is associated with a task. A vector element has different meanings according to the task type. For a nonpreemptive task, the integer element denotes the slot from which its task operation starts. A single number is enough to represent the task operation in the allocation table, as a nonpreemptive task goes to the end without being suspended, once it has started. It can start from its activation time to the latest start time, which can be calculated by subtracting U_i from D_i. In contrast, for preemptive tasks, the integer element means the binary map by which task operations are assigned to the time slots. Hence, the map tells how to select U_i, the length of the task operation, out of $(D_i - A_i + 1)$ slots, the length of the map. The number of choice options is bounded by M for nonpreemptive tasks while by $O(2^{M/2})$ for preemptive tasks [14].

Now, a fitness function evaluates a schedule according to the given system goal. For peak load estimation, the allocation table is converted from a chromosome. Then, we can calculate the power demand for each slot, and the maximum of them will be the peak load. The smaller the peak load, the better the schedule. In addition to this basic encoding scheme, the effect of delayed start must be further taken into account in the fitness function. It can be better described by an example shown in Figure 2. As shown in Figure 2(a), there are 4 tasks, so N is 4. T_0 and T_1 are nonpreemptive, while T_2 and T_3 are preemptive. In addition, we assume that the time window consists of 10 slots, making M equivalent to 10. T_0 is activated at the time slot 1 and must be completed by time slot 6. Its operation length is 3, so its profile entry has three numbers, each of which represents the power consumption on its time slots when the task is run.

Next, Figure 2(b) shows how a chromosome is converted to an allocation table. For an encoded vector of (2, 3, 58, 13), the element associated with T_0 is 2. It means that T_0 begins from time slot 2. Hence, the profile entry (3, 2, 3) is sequentially copied to T_0 row of the allocation table from time slot 2. The allocation for T_1 can be explained in the same way. For T_2, a preemptive task, its start time is 2 and deadline is 8. As its operation length is 4, 4 out of 7 slots need to be selected. In the vector, T_2 is associated with 58, and its binary equivalent is 0111010. The profile entry is mapped to the allocation table for each appearance of 1, from time slot 2. Hence, (0, 1, 2, 3, 0, 4, 0) is the allocation result for T_2, where the first 0 is the power demand in slot 2. Then, Figure 2(b) also shows the per-slot power demand just below the allocation table. The fitness function finds the maximum of them, so 9.0 is the fitness value for this allocation.

Figure 2(c) illustrates how to modify the allocation table by *P-Penalty* and *N-Penalty*. First, T_0 has *N-Penalty* of 4, and its start is delayed by 1 slot, while average per-slot power consumption is 2.67. Hence, $\lceil 1/4 \rceil$, namely, 1 extraslot is added, and the its power consumption is $2.67 \div 4 = 0.67$, as shown in time slot 5. Second, T_1 starts 3 slots later than its activation time, and its *N-Penalty* is 2. $\lceil 3/2 \rceil$, namely, 2 slots are added for this delay. They are slots 5 and 6. As average power consumption is 3.0, slot 5 has power consumption of 3 while slot 6 has 1.5. Next, T_2 has *P-Penalty* of 0.2, and it is delayed by 1 slot. Hence, power consumption in every slot is increased by 20%. For T_3, each slot increases by 10%. If a preemptive task has *N-Penalty* and p slots are additionally needed, p slots are selected out of slots from its start time (not its activation time) to deadline. The peak load takes place at the 5th slot, and peak value has changed to 10.6. It must be mentioned that slot length extension can lead to the

Task specification	Profile	*N-Penalty*	*P-Penalty*
T_0 (N, 1,3,6)	3, 2, 3	4	
T_1 (N, 0, 2, 8)	2, 4	2	
T_2 (P, 2, 4, 8)	1, 2, 3, 4		0.2
T_3 (P, 0, 3, 5)	3, 3, 3		0.05

(a) Task specification

Encoded vector: (2, 3, 58, 13)

	0	1	2	3	4	5	6	7	8	9
T_0			3	2	3					
T_1				2	4					
T_2				1	2	3		4		
T_3			3	3		3				
Per-slot	0	0	6	6	9	6	0	4	0	0

(b) Encoded schedule (allocation table)

	0	1	2	3	4	5	6	7	8	9
T_0			3	2	3	0.7				
T_1				2	4	3	1.5			
T_2				1.2	2.4	3.6		4.8		
T_3			3.3	3.3		3.3				
Per-slot	0	0	6.3	8.5	9.4	10.6	1.5	4.8	0	0

(c) After applying delayed start

Figure 2: Encoding and allocation table.

violation of deadline constraints for some tasks. However, those schedules will be eliminated in genetic iterations.

3.3. Genetic Iterations. With the design of an encoding scheme and the definition of the fitness function, genetic operations are executed, continuously improving the quality of population. The overall procedure is described in Figure 3. Each evolutionary step generates a population of candidate solutions and evaluates them according to a given fitness function. Even though the genetic algorithm can possibly fail to find an optimal solution, its efficiency makes itself very practical, as the schedule must be created within the system's tolerance. For initial population, a predefined number of chromosomes are generated randomly. In a chromosome, each element is selected from th e valid range of the associated task. That is all schedules in the population can meet the time constraint of each task. This restriction narrows the search scope and cuts off the chance of creating a better schedule by mating invalid schedules. However, without this restriction, the scheduler spends too much time in processing invalid schedules, which violate the time constraint of some tasks.

The iteration mainly consists of selection and reproduction. Selection is a method that picks parents according to the fitness function. The Roulette wheel selection gives more chances to chromosomes having better fitness values for mating. Actually, we have also tried other strategies such as the tournament selection; however, the Roulette wheel scheme performs better than others in most cases. Reproduction or crossover is the process of taking two parents and producing offspring with the hope that the offspring will be a better solution. This operation randomly selects a pair of two crossover points and swaps the substrings from each parent. Reproduction may generate the same chromosome with currently existing ones in the population. It is meaningless to have multiple instances of a single schedule, so they will be replaced by new random ones. Additionally, mutation exchanges two elements in a single chromosome. In our scheme, the meaning of the value is different for preemptive and nonpreemptive tasks. Hence, the mutation across the different task domains must be prohibited.

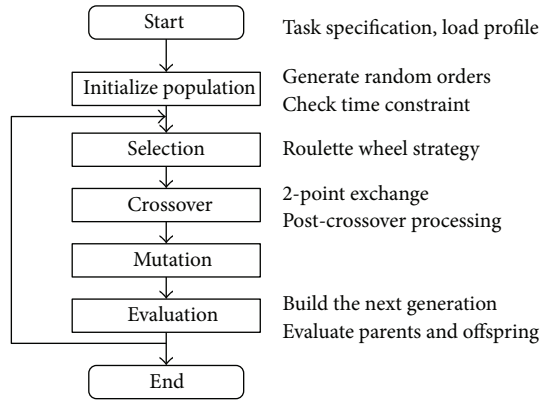


Figure 3: Scheduler operation.

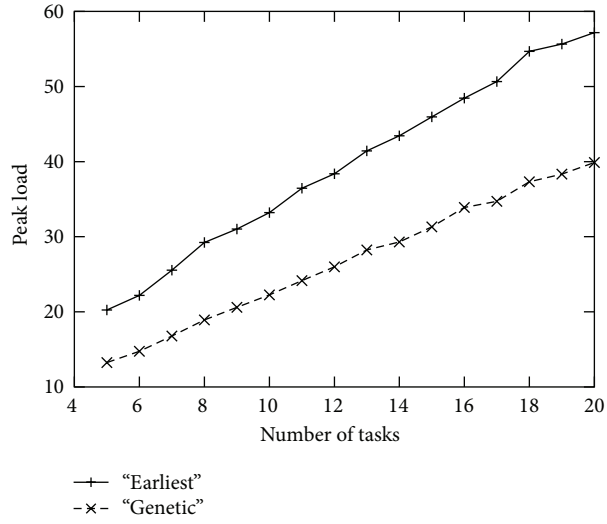
4. Performance Measurement

This section implements the proposed allocation method using Visual C++ 6.0, making it run on the platform equipped with Intel Core2 Duo CPU, 3.0 GB memory, and Windows Vista operating system. The time slot length is implicitly selected to be 5 min, considering the period of price signal change. For the scheduling window of 2 hours, M is set to 24. For a task, the start time is selected randomly between 0 and $M-1$, while the operation length and the slack exponentially distribute with the averages of 5.0 and 3.0 slots, respectively. A task will be discarded and replaced if the finish time, namely, the sum of start time and operation length, exceeds M. In addition, the power level for each time slot ranges from 1.0 to 10.0. The power scale is not explicitly specified, as it depends on the grid type and included devices. As for genetic operations, each population includes 96 chromosomes, while the number of iterations is set to 1,000. This configuration can create a schedule within 1 second. Actually, even with more iterations, the fitness value hardly gets improved.

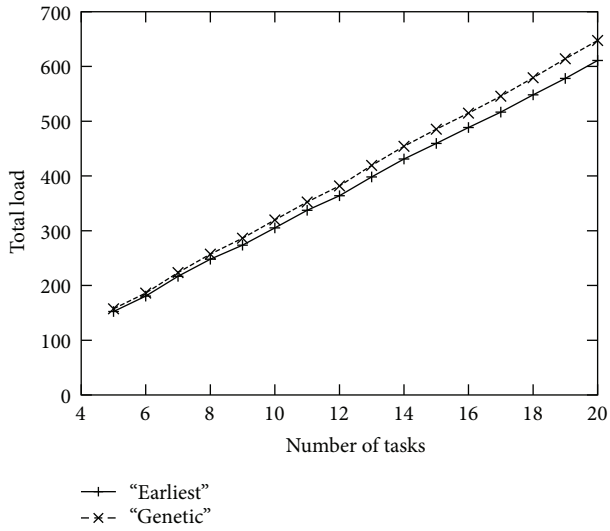
For performance comparison, the *Earliest* scheduling scheme initiates tasks as soon as they get ready and makes it run without preemption. It adopts no control strategy but provides a measure for a comparative assessment for the efficiency of other charging strategies. The experiment mainly focuses on peak load and total load for the schedule found by the proposed scheme. Extrapower load is not added to the *Earliest* scheme, as the activation time is the same as the start time for every task. Main performance metrics include the number of tasks, *P-Penalty*, *N-Penalty*, and slack. In the subsequent experiments, the number of preemptive tasks is set to 3, as the nonpreemptive tasks are more common. For each parameter setting, 30 sets are generated, and their results are averaged. In the performance comparison graphs, the curve marked by *Genetic* plots the performance of the proposed genetic scheduler.

The first experiment measures both peak load and total load according to the number of tasks, while the experiment results are plotted in Figure 4. Here, the number of tasks ranges from 4 to 20. For the experiment, every preemptive task is assumed to have *P-Penalty* of 0.05, while every nonpreemptive task is assumed to have *N-Penalty* of 3. As the average power consumption is almost the same for each task, for a sufficiently large number of tasks, both peak load and total load linearly increase along with the number of tasks. As shown in Figure 4(a), the proposed scheme reduces the peak load by 35.2% when the number of tasks is 8. The performance gap is generally uniform for the range of a given number of tasks. Total load indicates how much overhead is added due to peak load distribution. As shown in Figure 4(b), just a small difference is found between two cases of *Earliest* and proposed schemes. The difference tends to get larger when there are more tasks, but it remains less than 5%. This experiment discovers that our scheme achieves significant peak load reduction just with small increase in the total load.

The next experiment measures peak load and total load according to the *P-Penalty* ranging from 0 to 0.1, and the results are shown in Figure 5. In this experiment, the number of tasks is set to 10, and *N-Penalty* is to 3. Peak load and

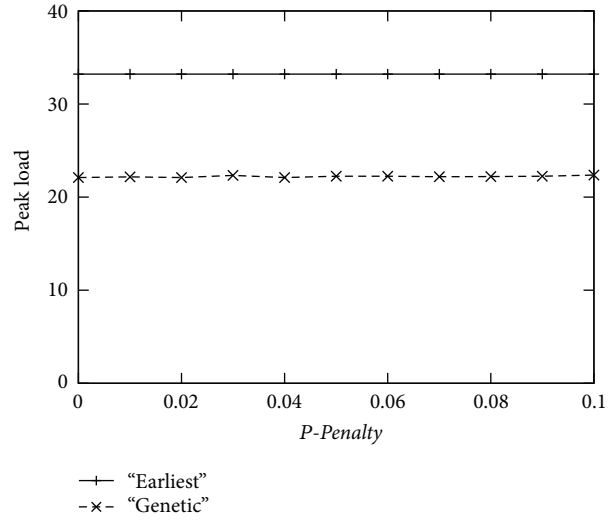


(a) Peak load

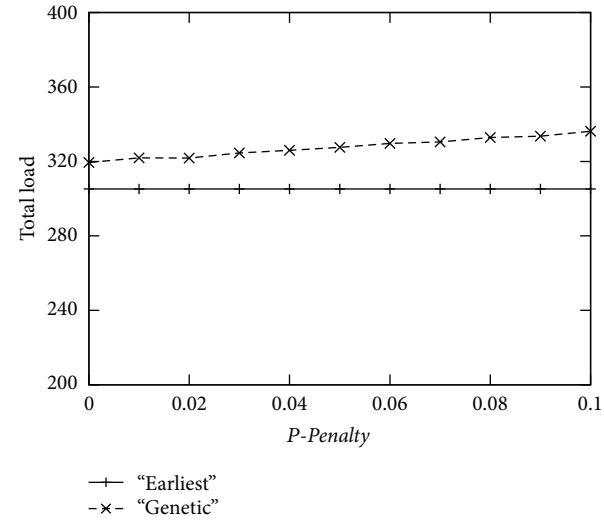


(b) Total load

Figure 4: Effect of the number of tasks.



(a) Peak load



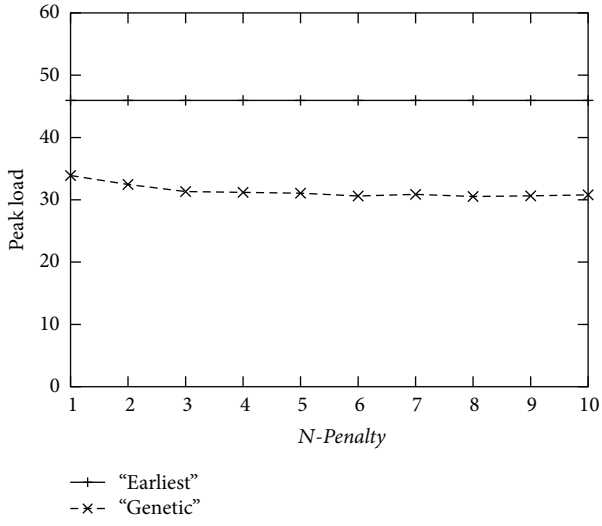
(b) Total load

Figure 5: Effect of *P-Penalty*.

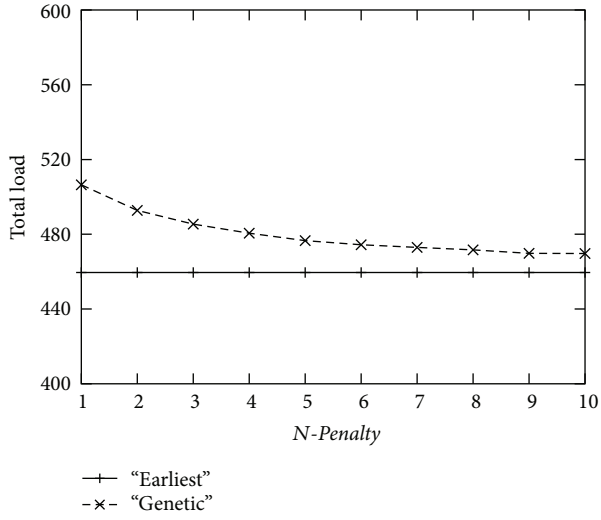
total load do not change in the *Earliest* scheme, as the number of tasks is fixed to 10. However, Figure 5(a) shows that peak load is rarely affected by the change of *P-Penalty* also in the proposed scheme. Peak load lies between 22.1 and 22.3. This result indicates that the proposed scheme can stably find an efficient schedule, even if peak load is expected to linearly increase along with *P-Penalty*. According to the graph, our genetic scheduler reduces peak load by around 33.1%, compared with the *Earliest* scheme. In addition, the total load increases when *P-Penalty* gets larger almost linearly as shown in Figure 5(b). The difference in the total load is about 10.1% between the two schemes when *P-Penalty* reaches 0.1.

In addition, Figure 6 traces the effect of *N-Penalty* to peak and total load, respectively. In the experiment, *N-Penalty* is changed from 1 to 10, while the number of tasks is set to 15 and *P-Penalty* to 0.05. A smaller value of *N-Penalty* makes the total power consumption increase more. If *N-Penalty* is 1, each slot of delay leads to one slot of extension. As in the previous experiment, both peak load and total load remain constant in the *Earliest* scheme, as the number of tasks is fixed. Our genetic scheduler places those tasks having severe penalty as earlier as possible to minimize the influence of delayed starts. As a result, the genetic scheduler reduces peak load by 26.2%, even when *N-Penalty* is 1. In addition, just 9.2% of load is added to the total power consumption, compared with the *Earliest* scheme.

Finally, Figure 7 measures the effect of task slack to peak load and total load, respectively. Here, the slack ranges from 1 to 8 slots, while the number of tasks is fixed to 15, *P-Penalty* to 0.05, and *N-Penalty* to 3. The larger the slack is, the more options scheduler has in placing task operation in the allocation table. Actually, the large number of options means the increase in the search space size. The genetic scheduler

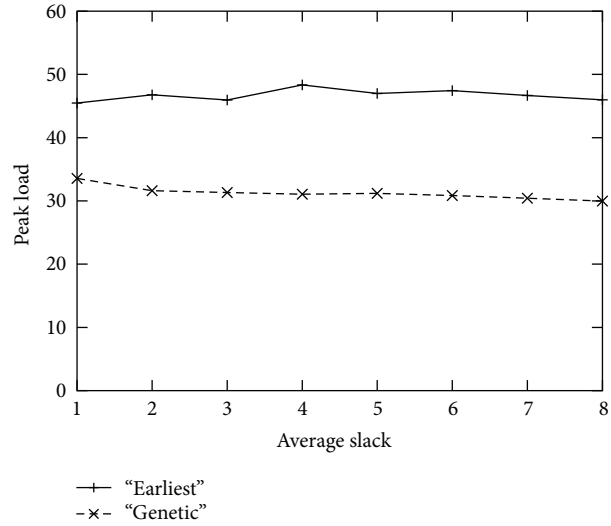


(a) Peak load

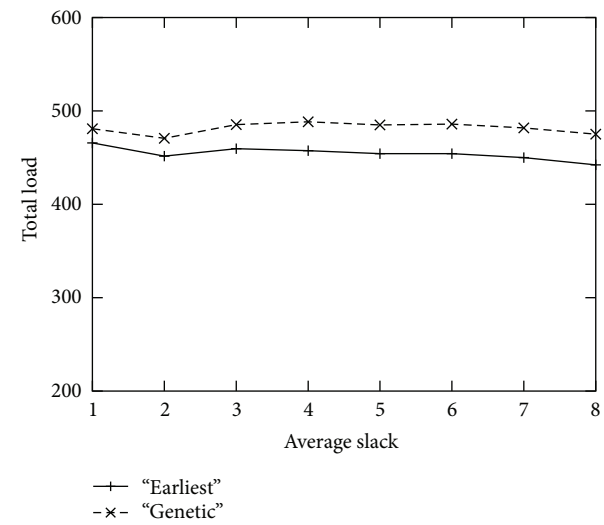


(b) Total load

Figure 6: Effect of *N-Penalty*.



(a) Peak load



(b) Total load

Figure 7: Effect of task slack.

has an additional tradeoff between the execution speed and accuracy. However, as the penalty also increases if the task operation is delayed for better distribution, it is difficult to design a straightforward policy in task scheduling. For the *Earliest* scheme, peak load is also affected by the slack size, even if not so much, as a smaller slack value is highly likely to make several tasks activated at the same time slot.

Figure 7(a) shows that peak load of the proposed scheme is smaller than that of the *Earliest* scheme by up to 35.7%, when the slack is 4 slots. The improvement is smallest when task operations are tight; namely, tasks have small slacks. Even though this experiment cannot find a regular improvement pattern due to two conflicting factors of even distribution and increased penalty, our scheme outperforms by at least 26.1% for the whole slack range. Additionally, according to Figure 7(b), total load gets larger according to the increase of the slack. The gap increases from 3.1% just to 6.8%. After all, this experiment discovers that our genetic scheduler takes advantage of more scheduling options stemmed from large slack, just with a little increase in the total load.

5. Conclusions

The smart grid is a future power system which essentially employs WSANs for its applications such as wireless meter reading and remote system monitoring. The power management for actuators gets more important, as many standard protocols such as Zigbee have already achieved significant improvement in sensor network part. This paper has designed an actuator operation scheduler capable of reducing peak load in power consumption of actuator tasks, taking advantage of genetic algorithms. Based on the load

profile specification and the task model consisting of nonpreemptive and preemptive tasks, each schedule is represented by a chromosome. Here, a schedule is also equivalent to an allocation time table. The fitness function calculates per-slot power consumption to find the peaking slot. In schedule evaluation, *P-Penalty* and *N-Penalty* account for the extended operation length or increased power consumption brought by the delayed start of some actuator operations, respectively.

Extensive experiments have been conducted to measure the performance of the proposed scheme mainly in terms of peak load and total load according to the number of tasks, *P-Penalty*, *N-Penalty*, and task slack. The experiment reveals that our genetic scheduler reduces the peak load by up to 35.2% for the given parameter set compared with the *Earliest* scheduling scheme. Moreover, genetic operations can find an efficient schedule, smartly compromising two conflicting objectives of even power load distribution and small actuator initiation delay. Our genetic scheduler can possibly integrate an intelligent heuristic in initial population selection and genetic loop customization. In addition, a new system goal can be defined such as cost reduction and renewable energy integration.

As future work, we are planning to integrate renewable energies and charging facilities for electric vehicles into our WSAN-based power management system for smart buildings. Future buildings are highly likely to install distributed power generation equipments for solar and wind energies, while electric vehicles put significant load on the power system if they are plugged in to the power grid simultaneously. Computational intelligence can overcome the complexity in coordinating many different power entities having their own roles in power consumption scenarios.

Acknowledgment

This paper was financially supported by the Ministry of Knowledge Economy (MKE), Korea Institute for Advancement of Technology (KIAT) through the Inter-ER Cooperation Projects.

References

[1] V. C. Gungor, B. Lu, and G. P. Hancke, "Opportunities and challenges of wireless sensor networks in smart grid," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 10, pp. 3557–3564, 2010.

[2] D. Gislason, *ZIGBEE Wireless Networking*, Newnes, Melbourne, Australia, 2008.

[3] Z. M. Fadlullah, M. M. Fouda, N. Kato, A. Takeuchi, N. Iwasaki, and Y. Nozaki, "Toward intelligent machine-to-machine communications in smart grid," *IEEE Communications Magazine*, vol. 49, no. 4, pp. 60–65, 2011.

[4] S. Bu, R. Yu, Y. Cai, and X. Liu, "When the smart grid meets energy-efficient communications: green wireless cellular networks powered by the smart grid," *IEEE Transactions on Wireless Communications*, vol. 11, no. 8, pp. 3014–3024, 2012.

[5] M. Erol-Kantarci and H. T. Mouftah, "Wireless sensor networks for cost-efficient residential energy management in the smart grid," *IEEE Transactions on Smart Grid*, vol. 2, no. 2, pp. 314–325, 2011.

[6] H. Doukas, K. D. Patlitzianas, K. Iatropoulos, and J. Psarras, "Intelligent building energy management system using rule sets," *Building and Environment*, vol. 42, no. 10, pp. 3562–3569, 2007.

[7] R. H. Katz, D. E. Culler, S. Sanders et al., "An information-centric energy infrastructure: the Berkeley view," *Sustainable Computing: Informatics and Systems*, vol. 1, no. 1, pp. 7–22, 2011.

[8] S. Shao, T. Zhang, M. Pipattanasomporn, and S. Rahman, "Impact of TOU rates on distribution load shapes in a smart grid with PHEV penetration," in *Proceedings of the IEEE PES Transmission and Distribution Conference and Exposition*, April 2010.

[9] T. Facchinetti, E. Bibi, and M. Bertogna, "Reducing the peak power through real-time scheduling techniques in cyber-physical energy systems," in *Proceedings of the 1st International Workshop on Energy Aware Design and Analysis of Cyber Physical Systems*, 2010.

[10] S. Sivanandam and S. Deepa, *Introduction to Genetic Algorithms*, Springer, Berlin, Germany, 2008.

[11] X. Cao, J. Chen, Y. Xiao, and Y. Sun, "Building-environment control with wireless sensor and actuator networks: centralized versus distributed," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 11, pp. 3596–3605, 2010.

[12] J. Taneja, D. Culler, and P. Dutta, "Towards cooperative grids: sensor/actuator networks for renewables integration," in *Proceedings of the 1st IEEE International Conference on Smart Grid Communications*, pp. 531–536, October 2010.

[13] J. Lee, H. Kim, G. Park, and H. Jeon, "Genetic algorithm-based charging task scheduler for electric vehicles in smart transportation," in *Intelligent Information and Database Systems*, vol. 7196 of *Lecture Notes in Artificial Intelligence*, pp. 208–217, Springer, 2012.

[14] J. Lee, H. Kim, G. Park, and M. Kang, "Energy consumption scheduler for demand response systems in the smart grid," *Journal of Information Science and Engineering*, vol. 28, pp. 955–969, 2012.

[15] J. Lee and G. Park, "Dual batterymanagement for renewable energy integration in EV charging stations," *NeuroComputing*. In press.