Enhanced Message-Passing Based LEACH Protocol for Wireless Sensor Networks

Jaeyoung Kang [1,†] , **Illsoo Sohn** [2,†] **and Sang Hyun Lee** [1,*,†]

1   School of Electrical Engineering, Korea University, Seoul 02841, Korea; retbird13@korea.ac.kr
2   Department of Computer Science and Engineering, Seoul National University of Science and Technology, Seoul 01811, Korea; isohn@seoultech.ac.kr
*   Correspondence: sanghyunlee@korea.ac.kr; Tel.: +82-2-3290-3218
†   These authors contributed equally to this work.

Abstract: This paper proposes a distributed energy-efficient clustering protocol for wireless sensor networks (WSNs). Based on low-energy adaptive clustering hierarchy (LEACH) protocol, the proposed LEACH-eXtended Message-Passing (LEACH-XMP) substantially improves a cluster formation algorithm, which is critical for WSN operations. Unlike the previous approaches, a realistic non-linear energy consumption model is considered, which renders the clustering optimization highly nonlinear and challenging. To this end, a state-of-the-art message-passing approach is introduced to develop an efficient distributed algorithm. The main benefits of the proposed technique are its inherent nature of a distributed algorithm and the saving of computational load imposed for each node. Thus, it proves useful for a practical deployment. In addition, the proposed algorithm rapidly converges to a very accurate solution within a few iterations. Simulation results ensure that the proposed LEACH-XMP maximizes the network lifetime and outperforms existing techniques consistently.

Keywords: LEACH; message passing; nonlinear power consumption

1. Introduction

Advances in sensor technology popularize battery-powered wireless sensor networks (WSNs) in a large number of industrial areas including vehicle traffic monitoring, smart factories, internet-of-things devices, and public safety networks to name a few [1,2]. As sophisticated functional features are necessary to achieve a high level of the automation for conveniences, additional wireless sensors are accommodated for covering an increasing demand of the sensing data. On top of a sophisticated network formation, sensors are subject to a variety of physical environments where the cost of intervening in the sensor network is massive [3,4]. Therefore, it is indispensable to extend the lifetime of the WSN.

There have been a number of efforts to maximize the longevity of the WSN, such as mobile relays, optimal deployment of sensor nodes, and energy harvesting [5]. In terms of routing and clustering, low-energy adaptive clustering hierarchy (LEACH) protocol has played a trailblazing role in the minimization of the energy consumption in the network [6]. This protocol uses a hierarchical clustering-based routing strategy which forms multiple clusters of nodes, each with a single cluster head (CH) node. In each group, member nodes transfer their information to the CH node, and the CH node is responsible for aggregating whole received information and forwarding it to the base station (BS). Besides single-hop clustering, there are several multi-hop hierarchical clustering based protocol. Power-efficient gathering in sensor information systems (PEGASIS) forms chains of sensor nodes such that each node transfers the data back and forth with adjacent nodes [7]. Hybrid energy-efficient distributed clustering (HEED) considers the residual energy and the proximity to adjacent nodes for the selection of the CH node [8]. In addition, energy-efficient uneven clustering (EEUC) protocol

partitions the set of sensor nodes into unequal sized clusters and uses multi-hop routing for inter-cluster communication to save the energy of the CH node located near the BS [9].

For simplicity of the LEACH protocol, several variations have been developed for the minimization of the network energy consumption. In LEACH-C [6], a central coordinator exists to control the cluster formation. By collecting all required network information, the central coordinator provides a solution for the optimized clusters. In LEACH-CE [10], a cluster head selection has been improved. Upon the formation of a cluster, the member node with the highest remaining energy is selected as a CH node. This enables to distribute the energy consumption load uniformly over the member nodes. In LEACH-CKM [11], a *K*-means (or *K*-medians) clustering algorithm has been adopted to enhance the clustering performance. With a sophisticated energy-efficient cluster formation, the WSN extends its lifetime.

However, aforementioned centralized protocols impose several practical requirements. They require accurate location information of all sensor nodes in the network. Additional hardware with localizing features, e.g., GPS, result in an increase in the cost and power consumption for sensor devices. Furthermore, the collection of the location information of individual nodes and the distribution of the clustering decision by the network coordinator become a signaling burden to the network. To cope with the practical issues, LEACH-AP has been proposed as a distributed solution [12]. LEACH-AP develops a distributed algorithm based on affinity propagation [13]. Hence, the location information of sensor nodes is not necessary and the signaling overhead becomes minimal. In addition, it outperforms the existing centralized counterparts in terms of the network lifetime. However, there are some drawbacks of LEACH-AP. LEACH-AP simplifies an energy consumption model of CH nodes for mathematical tractability [12]. The assumption that the sensing area is apart away from the BS and transmission energy dominates the energy consumption of CH nodes ignores the impact of the dynamic cluster size on the energy consumption of CH nodes in the simplified model. Thus, variable energy consumption for CH nodes has not been properly included, thereby resulting in an occasional deviation from the guaranteed optimality. In addition, LEACH-AP assumes that the sensing data within a cluster is closely correlated and thus all sensing data collected at the CH node are perfectly compressed. This assumption means that the size of the aggregated data at CH nodes is the same as the sensing data of each sensor node. In reality, the correlation between the sensing data of neighboring sensor nodes varies. The size of the aggregated data at CH nodes also differs depending on the number of data and their correlation, and may degrade the performance of the simplified model. However, revising the simplified energy consumption model of CH nodes to reflect a new feature renders the model highly nonlinear.

To resolve the issue, this work develops an energy consumption-minimizing strategy called LEACH-XMP (LEACH-eXtended Message-Passing) based on an enhanced technique of the message-passing algorithm [14]. Unlike the previous approach in [12], the LEACH-XMP adopts a sophisticated but realistic nonlinear energy consumption model for CH nodes without oversimplification. The resulting solution is very efficient in improving the cluster formation and in the lifetime maximization of whole WSN. The main contributions of this paper are summarized as follows:

- Generalized energy consumption model: This paper adopts a realistic energy consumption model of a sensor node. The model considers the transmission power dissipation and digital-processing power related to data decoding and aggregation. We capture the impact of the dynamic cluster size on the digital-processing power of a CH node which has been unaddressed in the previous studies. In addition, the data compression rate based on the correlation of the collected data is considered. It influences the transmission power dissipation of a CH node since an inaccurate energy consumption model of CH nodes incurs a suboptimal cluster formation. None of the previous works have properly addressed this issue to the best of the authors' knowledge.
- Distributed algorithm: The new energy consumption model renders the optimization formulation highly nonlinear. We propose a message-passing approach that can tackle this nonlinearity very efficiently. We construct a graphical model that corresponds to the WSN of interest and develop an efficient distributed algorithm based on it. The advantages of the proposed algorithm include that (i) the algorithm is distributed by nature; (ii) computational load for each node is

minimal; and (iii) the algorithm converges rapidly with only a small number of message-passing iteration. The advantages characterize the usefulness of the proposed algorithm for practical implementations.

- Performance: The performance of the proposed algorithm is verified with extensive simulation results. The proposed algorithm outperforms all the previous approaches in terms of energy consumption rate, total delivered data, and network lifetime. In particular, the proposed algorithm improves the network lifetime up to 28.5% compared to existing LEACH protocols in the network lifetime. Based on the realistic energy consumption model, the proposed algorithm provides an efficient solution for different network sizes. As the network size grows, the improvement also increases. This is one of the prominent main benefits of the proposed algorithm.

This paper is organized as follows: Section 2 describes the system model of generalized LEACH and Section 3 specifies the algorithm of LEACH-XMP. Subsequently, Section 4 provides simulation results of proposed algorithm. Finally, Section 5 draws the conclusion.

2. System Model

Consider a WSN that deploys a set of N sensor nodes denoted by $\mathcal{S}$ (see Figure 1). A single BS exists in the network and receives environmental sensing data from sensor nodes. Sensor nodes initially broadcast their identification number and remaining energy information. The received signal strength is used to estimate the distances from neighboring nodes. The LEACH protocol [6] proceeds with set-up phase and steady-state phase. In the set-up phase, clusters are constructed by grouping sensor nodes, where time-slot-based scheduling is also determined within each cluster. In the following steady-state phase, sensing packets from sensor nodes are collected at their CH nodes and then forwarded to the BS.
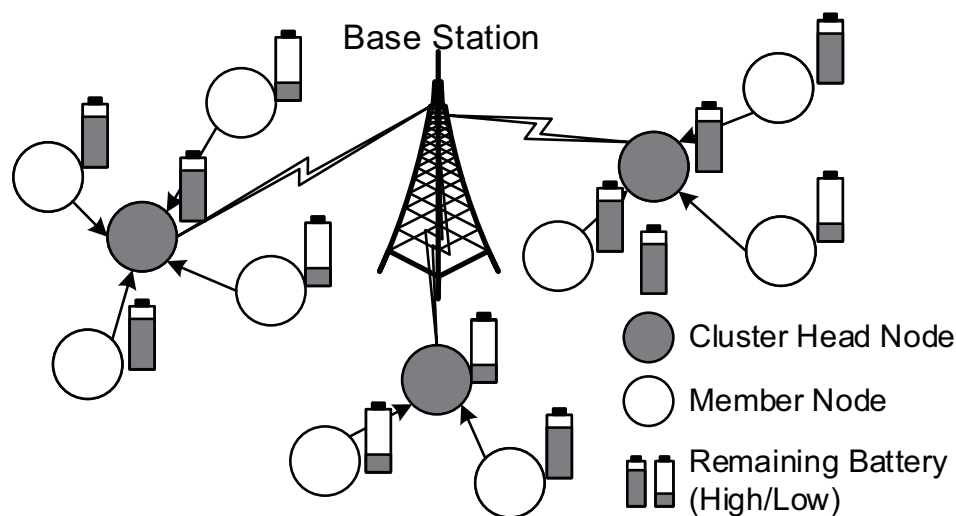

Figure 1. A battery-powered wireless sensor network (WSN) with low-energy adaptive clustering hierarchy (LEACH).

We desire to maximize the lifetime of the WSN. Such a goal is analogous to maximizing the remaining energy of the WSN. The remaining average total energy of sensors the WSN is monitored to maximize on a regular basis. Let τ_k be the k-th time instant where the measurement from sensors is reported to the BS. At time instant τ_k, all sensors measure their remaining energy along with the sensing data and share such information with neighbors. If at least one of their neighbors no longer work, they reconfigure the WSN to maximize the remaining energy. We assume that all sensors have the same amount of the initial energy denoted by $\mathcal{E}_0$. Let $\mathcal{E}_j^{(k)}$ and $\overline{\mathcal{E}}^{(k)}$ denote the remaining energy of sensor j and the average remaining energy of live nodes in the WSN at time instant τ_k, respectively. Thus, it holds that $\mathcal{E}_j^{(0)} = \mathcal{E}_0$ for all sensors. Also, the set of live sensors at time instant τ_k is denoted by

$\mathcal{S}_k$. Once the WSN is configured such that all sensors have their own roles in the network, sensor j consumes the energy $E_j^{(k-1)}$ depending on its role during the interval $[\tau_{k-1}, \tau_k]$. The remaining energy of sensor j is given by

$$\mathcal{E}_j^{(k)} = \mathcal{E}_j^{(k-1)} - E_j^{(k-1)} \tag{1}$$

and the the average remaining energy of live sensors is given by

$$\overline{\mathcal{E}}^{(k)} = \overline{\mathcal{E}}^{(k-1)} - \frac{1}{|\mathcal{S}_k|} \sum_{j \in \mathcal{S}_k} E_j^{(k-1)} \tag{2}$$

Hence, the maximization of the total remaining energy becomes equivalent to the minimization of the energy (corresponding to the second term in (2)) for the communication at each reporting interval.

The energy consumption of a sensor node consists of the transmission energy dissipation and the digital-processing energy consumption proportional to the amount of the processed data [6]. The transmission power dissipation depends on the amount of processed data and the propagation loss. To quantify this, we let E_{elec}, ϵ_{fs}, and ϵ_{mp} denote signal-processing energy per bit, transmission energy per bit in free-space propagation, and the transmission energy per bit in multipath propagation, respectively. The energy consumption of a single sensor varies from the wireless propagation model. According to the energy consumption model in [6], the free-space propagation becomes dominant when the transmission distance is small, while the multipath propagation prevails in the opposite case. A pairwise transmission is assumed to convey b bits of sensing data. The energy consumption of node i transmitting to node j is denote by w_{ij} and is given by

$$w_{ij} = \begin{cases} bE_{elec} + b\epsilon_{mp}d_{ij}^4 & \text{if } d_{ij} > d_0, \\ bE_{elec} + b\epsilon_{fs}d_{ij}^2 & \text{otherwise,} \end{cases} \tag{3}$$

where $d_0 \equiv \sqrt{\epsilon_{fs}/\epsilon_{mp}}$ is the threshold distance associated with the point where the energy consumptions by multipath and free-space propagation are identical. A CH node is responsible for processing and aggregating the received data into a single packet to transfer to the BS. In addition to the energy dissipation of E_{elec} per bit for the signal reception and transmission, a CH node consumes E_{da} per bit to aggregate own received data. For an n-member cluster, the CH node transmits the collected information of nb bits to the BS. In previous works [6], it is assumed that the CH node can compress the collected information of nb bits into a single data packet of a constant length b to transmit to the BS. However, in practice, the aggregated information may not be compressed into a sufficiently compact data packet. The number of bits required for representing the information can increase according to the relationship among the data from distinct nodes. To characterize it, we consider a generalized model by considering the correlation parameter between different data, denoted by λ. This parameter, taking on a value from $[0, 1]$, is, in fact, associated with the best rate which an individual member signal can be compressed with. Let b_n denote the number of bits that the CH node of an n-member cluster can obtain by compressing n different data of b bits. Under the assumption of uniform λ among all data from individual sensors, b_n is evaluated by the inclusion-exclusion principle as a nonlinear function of n, which can be expressed as

$$b_n = \frac{1 - (1 - \lambda)^n}{\lambda} b \tag{4}$$

Note that, if λ is zero, i.e., all measurements are independent of each other and share no common information, b_n becomes nb. On the other hand, if λ is one, i.e., all measurements are identical and only one measurement is enough, b_n becomes b, which corresponds to the assumption made in previous works [6]. This function increases sublinearly with respect to n but decreasing with respect to λ as

depicted in Figure 2. Thus, the transmitting energy cost per bit of CH node j which is apart from the BS by the distance d_j amounts to

$$w_j = \begin{cases} E_{elec} + \epsilon_{mp}d_j^4 & \text{if } d_j > d_0, \\ E_{elec} + \epsilon_{fs}d_j^2 & \text{otherwise.} \end{cases} \tag{5}$$

We include additional requirement for the eligibility of a CH node [12]. Only a node that holds $\mathcal{E}_j^{(k)} \geq \overline{\mathcal{E}}^{(k)}$ is eligible for the CH selection. Here, $\overline{\mathcal{E}}^{(k)}$ can be acquired by consensus-based techniques [15] without global notification of the state of individual nodes.
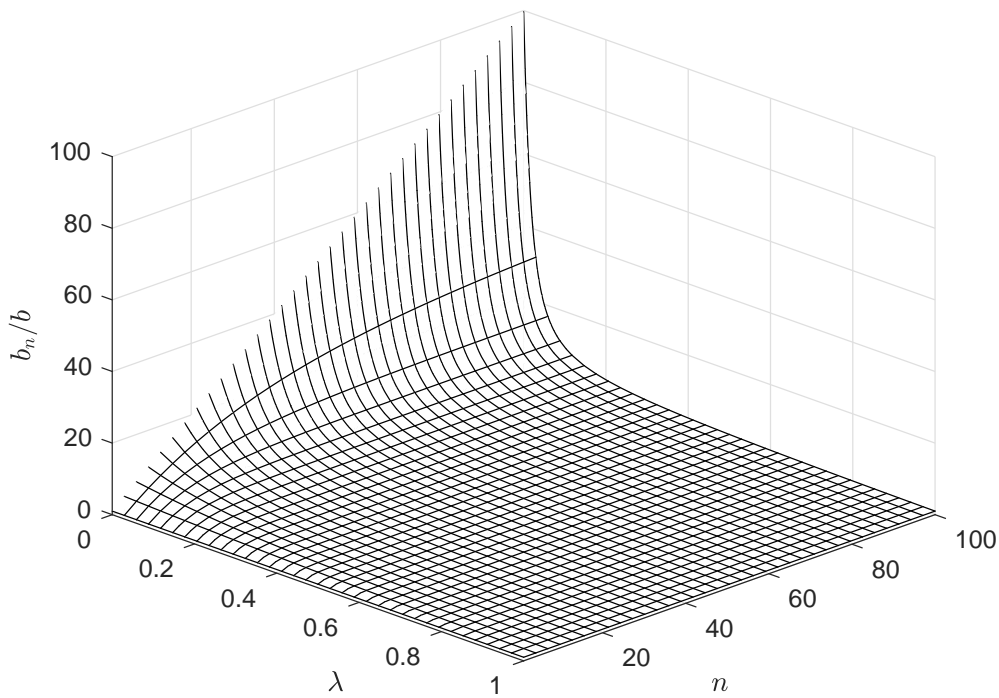

Figure 2. Representation of b_n/b for varying λ and n.

For clustering formation, several decision tasks are fulfilled such that the number of clusters and their CH nodes are identified and member nodes are affiliated to one of those clusters. Each node decides its own role between CH and member node by considering the corresponding energy consumptions as similarity measures used in the clustering problem. Also, once a node is chosen as a CH node, it cannot become a member node of another cluster. These decisions boil down to two requirements: (i) Each node chooses its CH node. To become a CH node, the node should choose itself as its CH node; (ii) once a node is chosen by other nodes to act as a CH node, it cannot choose another node as its CH node. For concrete mathematical formulation, we introduce binary variable x_{ij} to indicate the membership of node i, such that $x_{ij} = 1$ indicates node i selects node j as a CH node. Once node i selects node j as its CH node, node j should choose itself as own CH node, i.e., $x_{jj} = 1$. Let $\mathcal{N}(j)$ and u_j be the set of neighboring nodes around node j and the number of neighboring nodes connected to CH node j (excluding CH node j), respectively. It holds that $u_j = \sum_{i \in \mathcal{N}(j)} x_{ij}$. We can reformulate the energy consumption of CH node j as

$$f(x_{jj}, u_j) = \begin{cases} 0 & \text{if } x_{jj} = 0, \\ bw_j & \text{else if } u_j = 0, \\ u_j bE_{elec} + (u_j + 1)bE_{da} + b_{u_j+1}w_j & \text{else if } u_j > 0. \end{cases} \tag{6}$$

Note that unlike the simplified model in [12], Equation (6) captures a nonlinear energy consumption model of a CH node. The resulting formulation of the WSN is given by

$$\min_{\{x_{ij}, u_j\}} \sum_{i \in \mathcal{S}_k} \sum_{j \in \mathcal{N}(i)} w_{ij} x_{ij} + \sum_{j \in \mathcal{S}_k} f(x_{jj}, u_j) \tag{7a}$$

$$\text{subject to} \sum_{j \in \mathcal{N}(i)} x_{ij} = 1, \ \forall i \tag{7b}$$

$$\sum_{i \in \mathcal{N}(j)} x_{ij} = u_j x_{jj}, \ \forall j \tag{7c}$$

$$x_{ij} \in \{0, 1\}, \ u_j \in \{0, \ldots, |\mathcal{N}(j)|\}, \tag{7d}$$

where $|\mathcal{N}(j)|$ is the number of neighbors around node j. The first term of the objective function in (7a) corresponds to the energy consumption of member nodes and the second term corresponds to the energy consumption of CH nodes. The first constraint indicates that each member can select only one CH node and the second constraint dictates that the selected node chooses itself as a CH node of the affiliated cluster. Note that solving the optimization in (7a)–(7d) is a computationally demanding task since the problem is inherently of combinatorial nature in addition to high nonlinearity of objective function and constraints.

3. Proposed LEACH-XMP Protocol

3.1. Formulation

To tackle the optimization problem in (7a)–(7d), we use a message-passing algorithm which has had remarkable success in various applications [12–14]. For the development of a distributed algorithm, Equations (7a)–(7d) is reformulated to an unconstrained optimization as

$$\min_{\{x_{ij}\}} \mathcal{F}(x_{ij}) \equiv \sum_{i \in \mathcal{S}_k} \sum_{j \in \mathcal{N}(i)} S_{ij}(x_{ij}) + \sum_{i \in \mathcal{S}_k} Q_i(\{x_{ij}\}) + \sum_{j \in \mathcal{S}_k} R_j(\{x_{ij}\}) \tag{8}$$

where new functions are defined to enforce the constraints as

$$S_{ij}(x_{ij}) = w_{ij} x_{ij} \tag{9}$$

$$Q_i(\{x_{ij}\}) = \begin{cases} \infty & \text{if } \sum_{j \in \mathcal{N}(i)} x_{ij} \neq 1 \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

$$R_j(\{x_{ij}\}) = \begin{cases} \infty & \text{if } \mathcal{E}_j^{(k)} < \overline{\mathcal{E}}^{(k)} \text{ or } \sum_{i \in \mathcal{N}(j)} x_{ij} \neq u_j x_{jj} \\ f(x_{jj}, u_j) & \text{otherwise} \end{cases} \tag{11}$$

where $S_{ij}(x_{ij})$ is introduced to represent the individual contribution of a member node to the total energy consumption of the WSN in (7a), and $Q_i(\{x_{ij}\})$ is associated with the constraint (7b) that each sensor node selects its own single CH node. To see this, if node i chooses more than two neighboring nodes as its CH node, i.e., there exists more than $j \in \mathcal{N}(i)$ such that $x_{ij} = 1$, the value of $Q_i(\{x_{ij}\})$ becomes infinity, and the objective function in (8) is never minimized. Otherwise, $Q_i(\{x_{ij}\})$ takes the value of zero and does not contribute to the final objective function. Thus, the value of (8) becomes identical to the objective of the original problem in (7a)–(7d) when minimized. Also, $R_j(\{x_{ij}\})$ represents the constraint for a CH node in (7c) and its energy consumption. If the corresponding constraint is violated, the value of $R_j(\{x_{ij}\})$ becomes infinity and the minimization does not work. Otherwise, $R_j(\{x_{ij}\})$ takes the value of $f(x_{jj}, u_j)$ which contributes to the total energy consumption of the network. Note here that $R_j(\{x_{ij}\})$ is a function of $\{x_{ij}\}$ solely since u_j is thought of as a function of $\{x_{ij}\}$. In turn, a graphical model associated with (8) is constructed. The graphical model is a bipartite graph consisting of two groups of nodes associated with variables (circles) and functions (boxes) interconnected by the set

of edges, each representing the membership of a variable in the corresponding function. Figure 3a shows a graphical model of 3-node WSN. A message-passing algorithm exchanges real-valued quantities called messages along two directions of each edge in the graphical model to obtain a distributed solution. Since there are three different types of function (9)–(11), six types of messages could be defined. However, only a single variable is connected with (9) and the corresponding message remains a trivial constant. Thus, four different messages (as illustrated in Figure 3b) suffice to obtain the message-passing algorithm. A single iteration consists in updating each message along edges between function and variable nodes. This iteration is repeated until all message converge. The converged messages are associated with the preference of the value that each variable can take in the optimal solution i.e., if a message is positive, the corresponding binary variable becomes one in the optimal solution. Figure 3c shows an example of the result of message-passing algorithm upon convergence. The sum of a pair of messages transferred in the opposite directions determines the value of a variable, which is associated with the CH-member node relationship. Note that this example corresponds to the case where node 1 is a CH node of the cluster that contains node 3, while node 2 stands alone.

3.2. Message Updates

The message-passing algorithm finds the solution of the unconstrained formulation (8). The algorithm runs in such an iterative fashion that two real-valued messages are transferred between a variable and a factor in two reciprocal directions of the in-between edge. The iterations of the message transfer are repeated until all messages converge to a fixed assignment of variables. Upon convergence of messages, the associated solution has been proved to be optimal in various applications.

Let $\mu_{A \to B}(C = c)$ denote a message that node A transfers node B to convey the chance of variable C taking the value of c. The information contained in the message can be thought of as the preference that optimal solution takes variable C equal to c. For convenience of representation, four different messages, respectively associated with two directions of edges connected to two factor functions, are reexpressed as:

$$\begin{aligned}
\beta_{ij} &= \mu_{x_{ij} \to Q_i}(x_{ij} = 1) - \mu_{x_{ij} \to Q_i}(x_{ij} = 0),\\
\eta_{ij} &= \mu_{Q_i \to x_{ij}}(x_{ij} = 1) - \mu_{Q_i \to x_{ij}}(x_{ij} = 0),\\
\rho_{ij} &= \mu_{x_{ij} \to R_i}(x_{ij} = 1) - \mu_{x_{ij} \to R_i}(x_{ij} = 0),\\
\alpha_{ij} &= \mu_{R_i \to x_{ij}}(x_{ij} = 1) - \mu_{R_i \to x_{ij}}(x_{ij} = 0).
\end{aligned} \tag{12}$$

We first consider messages that sent from variable node to constraint node. According to [14], β_{ij} can be calculated by the sum of all incoming messages to the variable node and is given by

$$\beta_{ij} = \begin{cases} \alpha_{ij} + w_{ij} & \text{if } i \neq j,\\ \alpha_{ii} & \text{otherwise.} \end{cases} \tag{13}$$

Also, η_{ij} can also be obtained in the same way as β_{ij} and is given by

$$\rho_{ij} = \begin{cases} \eta_{ij} + w_{ij} & \text{if } i \neq j,\\ \eta_{ii} & \text{otherwise.} \end{cases} \tag{14}$$
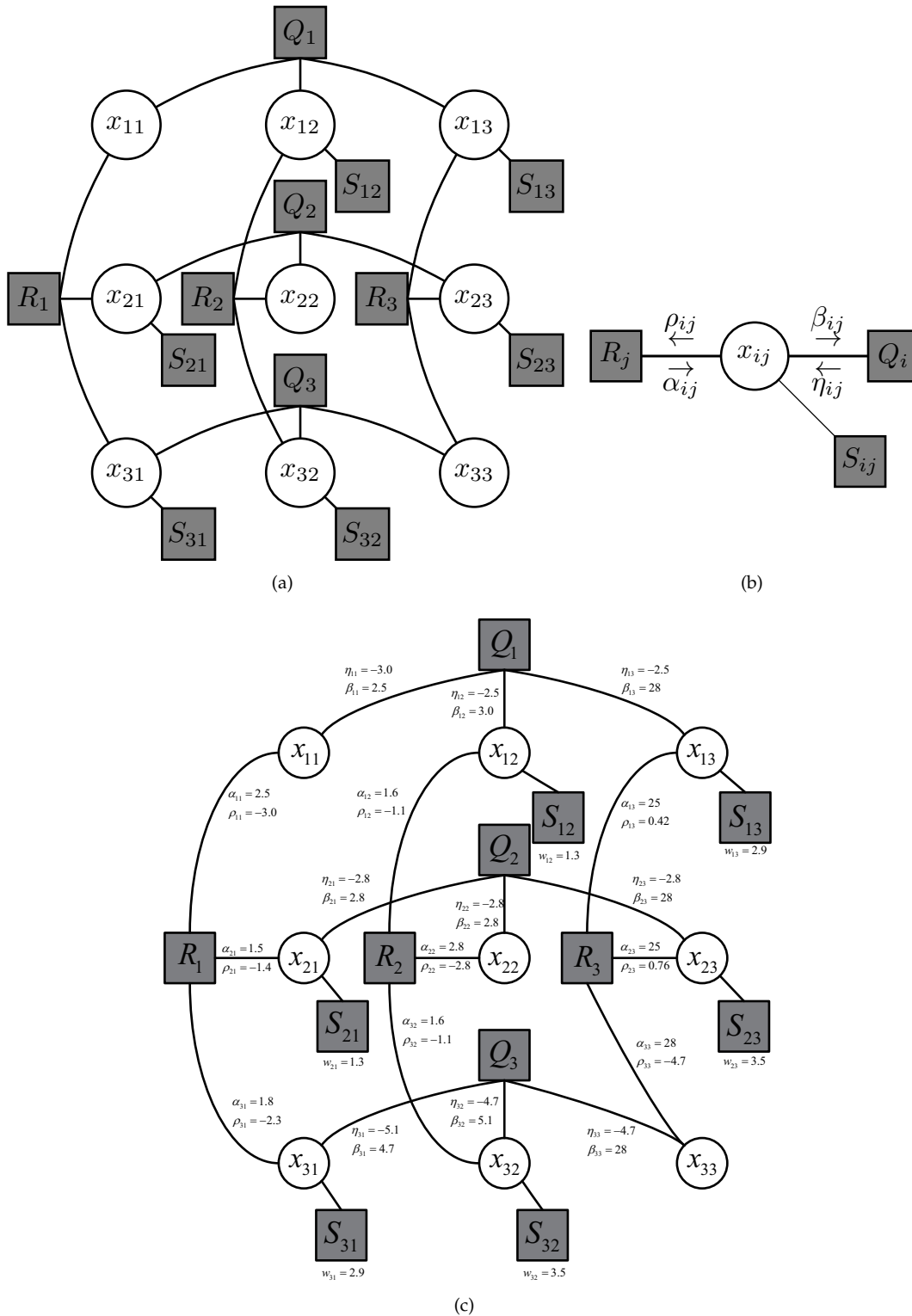
Figure 3. Graphical models. (**a**) Graphical model of 3-node WSN; (**b**) Message definition around a single variable; (**c**) Example of message-passing operation for 3-node WSN.

The message update rules for messages sent from a constraint node to a variable node are much complicated. The message sent from x_{ij} to function $Q_i(\{x_{ij}\})$ is considered first. Note that only one variable associated with function $Q_i(\{x_{ij}\})$ can take the value of one. If all incoming messages from other variables are negative, the outgoing message should be positive. By some algebra, η_{ij} is obtained in a simple form as

$$
\begin{aligned}
\eta_{ij} =\ & \mu_{Q_i \to x_{ij}}(x_{ij}=1) - \mu_{Q_i \to x_{ij}}(x_{ia}=0) \\
=\ & \min_{\mathcal{X}_i \backslash x_{ij}} Q_i(x_{ij}=1, \mathcal{X}_i \backslash x_{ij}) + \sum_{k \in \mathcal{N}(i) \backslash j} \mu_{x_{ik} \to Q_i}(x_{ik}) - \min_{\mathcal{X}_i \backslash x_{ij}} Q_i(x_{ij}=0, \mathcal{X}_i \backslash x_{ij}) + \sum_{k \in \mathcal{N}(i) \backslash j} \mu_{x_{ik} \to Q_i}(x_{ik}) \\
=\ & \sum_{k \in \mathcal{N}(i) \backslash j} \mu_{x_{ik} \to Q_i}(0) - \min_{k \in \mathcal{N}(i) \backslash j} \left(\mu_{x_{ik} \to Q_i}(1) + \sum_{l \notin \mathcal{N}(i) \backslash \{j,k\}} \mu_{x_{il} \to Q_i}(0) \right) \\
=\ & -\min_{k \in \mathcal{N}(i) \backslash j} \left(\mu_{x_{ik} \to Q_i}(1) - \mu_{x_{ik} \to Q_i}(0) \right) \\
=\ & -\min_{k \in \mathcal{N}(i) \backslash j} \beta_{ik}.
\end{aligned}
\tag{15}
$$

For the sake of simple representation, all three message rules in (13)–(15) are combined to obtain a single message update rule given by

$$
\rho_{ij} = w_{ij} - \min_{k \in \mathcal{N}(i) \backslash j} (w_{ik} + \alpha_{ik})
\tag{16}
$$

Next, the message update rule for α_{ij} is presented. To obtain α_{ij}, function $R_j(\{x_{ij}\})$ is minimized for two cases of $x_{ij}=1$ and $x_{ij}=0$, and the difference of their results is calculated. The resulting message update rule varies according to the relationship between i and j. If $i \neq j$, the corresponding message is obtained as

$$
\begin{aligned}
\alpha_{ij} =\ & \mu_{R_j \to x_{ij}}(x_{ij}=1) - \mu_{R_j \to x_{ij}}(x_{ia}=0) \\
=\ & \min_{\mathcal{X}_j \backslash x_{ij}} R_j(x_{ij}=1, \mathcal{X}_j \backslash x_{ij}) + \sum_{k \in \mathcal{N}(j) \backslash i} \mu_{x_{kj} \to R_j}(x_{kj}) - \min_{\mathcal{X}_j \backslash x_{ij}} R_j(x_{ij}=0, \mathcal{X}_j \backslash x_{ij}) + \sum_{k \in \mathcal{N}(j) \backslash i} \mu_{x_{kj} \to R_j}(x_{kj}) \\
=\ & \min[f(1,1) + \rho_{jj}, f(1,2) + \rho_{jj} + \min_{k \neq i} \rho_{kj}, f(1,3) + \rho_{jj} + \sum_{\substack{l=1 \\ k \neq i}}^{2} \mathrm{rnk}(l)\rho_{kj}, \ldots, \\
& f(1, N-1) + \rho_{jj} + \sum_{\substack{l=1 \\ k \neq i}}^{N-2} \mathrm{rnk}(l)\rho_{kj}] \\
& - \min[f(0,0), f(1,0) + \rho_{jj}, f(1,1) + \rho_{jj} + \min_{k \neq i} \rho_{kj}, f(1,2) + \rho_{jj} + \sum_{\substack{l=1 \\ k \neq i}}^{2} \mathrm{rnk}(l)\rho_{kj}, \ldots, \\
& f(1, N-2) + \rho_{jj} + \sum_{\substack{l=1 \\ k \neq i}}^{N-2} \mathrm{rnk}(l)\rho_{kj}] \\
=\ & \min[f(1,1), f(1,2) + \min_{k \neq i} \rho_{kj}, f(1,3) + \sum_{\substack{l=1 \\ k \neq i}}^{2} \mathrm{rnk}(l)\rho_{kj}, \ldots, f(1, N-1) + \sum_{\substack{l=1 \\ k \neq i}}^{N-2} \mathrm{rnk}(l)\rho_{kj}] \\
& - \min[-\rho_{jj}, f(1,0), f(1,1) + \min_{k \neq i} \rho_{kj}, f(1,2) + \sum_{\substack{l=1 \\ k \neq i}}^{2} \mathrm{rnk}(l)\rho_{kj}, \ldots, f(1, N-2) + \sum_{\substack{l=1 \\ k \neq i}}^{N-2} \mathrm{rnk}(l)\rho_{kj}], \\
=\ & \mathcal{A}_{ij}(1, N-1) - \min[-\rho_{jj}, \mathcal{A}_{ij}(0, N-1)],
\end{aligned}
\tag{17}
$$

where $\mathrm{rnk}(l)$ denotes the lth smallest input and $\mathcal{A}_{ij}(x,y)$ is defined with respect to message set $\{\rho_{kj} : k \neq i\}$ as

$$
\mathcal{A}_{ij}(x,y) = \min[f(1,x), f(1, x+1) + \min_{k \neq i} \rho_{kj}, f(1, x+2) + \sum_{\substack{l=1 \\ k \neq i}}^{2} \mathrm{rnk}(l)\rho_{kj}, \ldots
$$
$$
f(1, x+y-1) + \sum_{\substack{l=1 \\ k \neq i}}^{y-1} \mathrm{rnk}(l)\rho_{kj}]
\tag{18}
$$

On the other hand, if $i = j$, the message update rule can be derived in the similar way as

$$
\alpha_{jj} = \mathcal{A}_j(0, N)
\tag{19}
$$

where $\mathcal{A}_j(x,y)$ is defined similarly with respect to message set $\{\rho_{kj} : \forall k\}$ as

$$
\mathcal{A}_j(x,y) = \min[f(1,x), f(1, x+1) + \min_{i} \rho_{ij}, f(1, x+2) + \sum_{\substack{l=1 \\ i}}^{2} \mathrm{rnk}(l)\rho_{ij}, \ldots
$$
$$
f(1, x+y-1) + \sum_{\substack{l=1 \\ i}}^{y-1} \mathrm{rnk}(l)\rho_{ij}]
\tag{20}
$$

Since message ρ_{ij} of (16) is a function of α_{ij} and vice versa, the resulting algorithm is obtained as two self-consistent update rules. To handle the self-consistency, the algorithm proceeds in an iterative way using the following equations:

$$\alpha_{ij}^{(t)} = \begin{cases} \mathcal{A}_{ij}^{(t)}(1, N-1) - \min[-\rho_{jj}^{(t)}, \mathcal{A}_{ij}^{(t)}(0, N-1)] & \text{if } i \neq j \\ \mathcal{A}_j^{(t)}(0, N) & \text{otherwise} \end{cases} \tag{21}$$

$$\rho_{ij}^{(t+1)} = w_{ij} - \min_{k \in \mathcal{N}(i) \setminus j} (w_{ik} + \alpha_{ik}^{(t)}) \tag{22}$$

Note that, for physical deployment, member node i carries out the message calculations associated with $Q_i(\{x_{ij}\})$ and $S_{ij}(\{x_{ij}\})$ internally and combines them to prepare the outgoing message ρ_{ij} to CH candidate node j. Thus, Then, α_{ij} and ρ_{ij} message are sent to node j. Thus, only two types of messages α_{ij} and ρ_{ij} suffice for the message-passing iteration.

3.3. Cluster Head Selection

Node i updates messages $\alpha_{ij}^{(t)}$ and $\rho_{ij}^{(t)}$ repeatedly at each iteration and determine its preferred state via $\mathcal{U}_i = \min_{j \in \mathcal{N}(i)} [\rho_{ij}^{(t)} + \alpha_{ij}^{(t)}]$. The iteration continues until the iteration count reaches the maximally allowed number t_{max}, or $\mathcal{U}$ changes below a predetermined threshold σ. On convergence, node i determines its CH using $\mathcal{I}$ by

$$\mathcal{I}_i = \arg \min_{j \in \mathcal{N}(i)} [\rho_{ij}^{(t)} + \alpha_{ij}^{(t)}] \tag{23}$$

If $\mathcal{I}_i = i$, node i becomes a CH node and forms a new cluster.

A brief consideration of the computational complexity is ensured here. At a single iteration, most of computational efforts are made in calculating messages of $\alpha_{ij}^{(t)}$, which requires as many operations as $O(N \log N)$ for a single node in sorting input messages. Thus, it is expected that the worst case complexity for a single node amounts to $O(t_{max} N \log N)$ with the maximally allowed iteration number t_{max}. Since there are N nodes in the network, the overall computational complexity amounts to $O(t_{max} N^2 \log N)$. The overall algorithm is presented in Algorithm 1.

Algorithm 1 Cluster formation algorithm

For each (i, j), initialize $t \leftarrow 0$, $\rho_{ij}^{(t)} \leftarrow 0$, and $\alpha_{ij}^{(t)} \leftarrow 0$.

repeat

    Update $\alpha_{ij}^{(t)}$ using (21) and send to neighbors.
    Update $\rho_{ij}^{(t+1)}$ using (22) and send to neighbors.
    Increment $t \leftarrow t + 1$
until $|\mathcal{U}_i^{(t+1)} - \mathcal{U}_i^{(t)}| < \sigma$ for all i or $t > t_{max}$.
Node i chooses $\mathcal{I}_i$ as its CH using (23).

---

4. Simulation Results

The simulation is conducted using standard Python 3 with NumPy and SciPy libraries. In this simulation, 100 sensor nodes are initially scattered at random in a 100 m × 100 m square region in between $(0, 0)$ and $(100, 100)$. Simulation parameters are set referring to [6] and summarized Table 1. We refer to the interval between consecutive cluster reformations as a single round, which amounts to 7 days. The predetermined cluster number required by the existing algorithms for comparison is set to $K = 5$ [6]. Results are averaged over 100 random drops of wireless sensors. The lifetime of the WSN is calculated by the number of live nodes, which is evaluated at each round that consists of the set-up phase and the steady-state phase. The network lifetime and total amount of transferred data are measured by evaluating the number of rounds until the death of the last node.

Table 1. Simulation Parameters.

Description	Value
Radio circuitry energy dissipation, E_{elec}	50 nJ/bit
Energy dissipation of amplifier in free-space, ϵ_{fs}	10 pJ/bit/m^2
Energy dissipation of amplifier in multipath, ϵ_{mp}	0.0013 pJ/bit/m^4
Energy consumption for data aggregation, E_{da}	5 nJ/bit/signal
Initial energy of each sensors, $\mathcal{E}_0$	2 J
Initial number of sensors	100
Reporting interval	1 h
Cluster reformation interval	7 days
Packet size (/node/report)	200 bytes
Maximum value of allowed iterations, t_{max}	50
Stopping threshold, σ	0.001

We compare the performance of the proposed LEACH-XMP with existing LEACH-based protocols including LEACH-C, LEACH-CE, LEACH-CKM, and LEACH-AP. Since the existing techniques assume that the collected data can be perfectly compressed at CH nodes, i.e., $\lambda = 1$ for fair comparison, which is the special case of the proposed LEACH-XMP. Figure 4 compares the ratio of live nodes over time, while Figure 5 compares the total amount of data received at the BS over time in case A where the BS is located at (50, 175). As time elapses, the number of live nodes decreases. LEACH-XMP provides 18.7%, 28.5%, 21.5%, and 5.33% increase in the network lifetime as compared to LEACH-C, LEACH-CE, LEACH-CKM, and LEACH-AP, respectively. Accordingly, LEACH-XMP transfers the largest amount of sensing data to the BS among the LEACH variations.
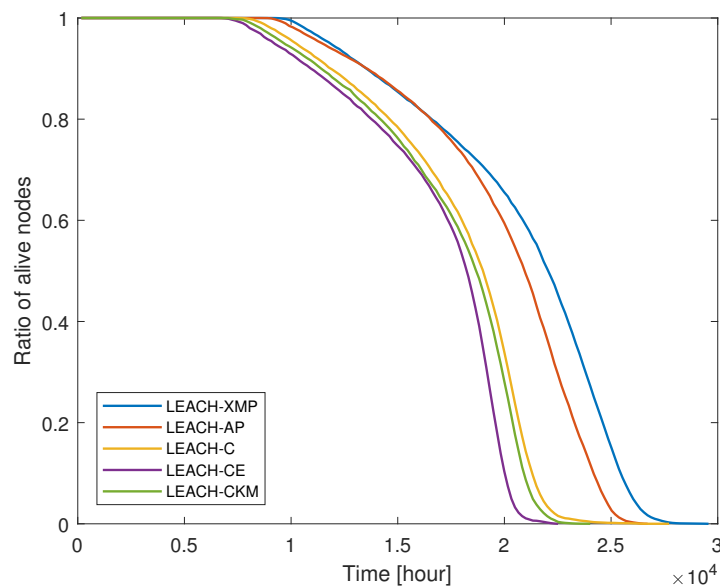
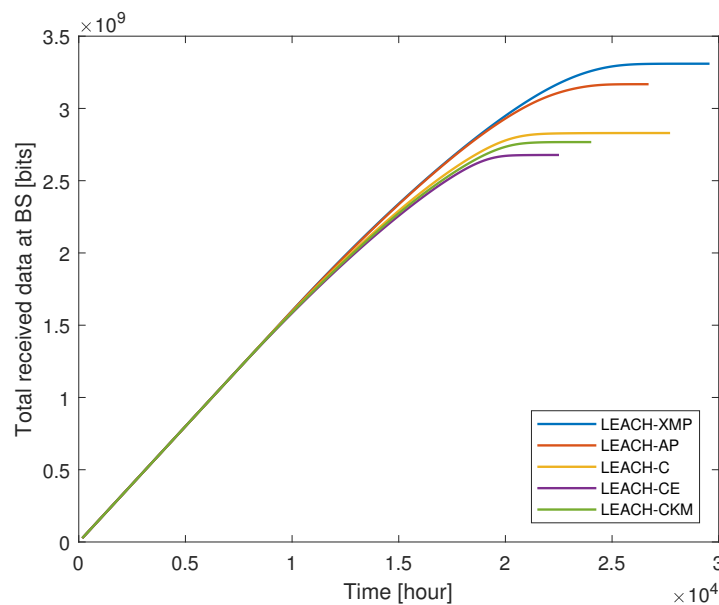**Figure 4.** Ratio of live nodes versus time.

Figure 5. Total amount of data received at the base station (BS) versus time.

Table 2 compares the computational complexity of LEACH-XMP and existing LEACH variations. In LEACH-C and LEACH-CE, the cluster formation is fulfilled by a simulated annealing algorithm [6]. At each iteration of the simulated annealing algorithm, the roles of the partial set of nodes are interchanged to improve a temporal solution, and the resulting computational complexity per iteration is low, i.e., $O(tKN)$. However, a simulated annealing algorithm typically requires at least thousands of iterations for convergence, t. Specifically, we set $t = 60,000$ in our simulations considering its slow convergence. On the other hand, our simulations results show that LEACH-CKM requires relatively small number of iterations, e.g., less than 10 iterations in most cases. In general, the performance of the K-medians algorithm adopted in LEACH-CKM is strongly affected by the initial choice of the value of K and CH nodes. Thus, the algorithm requires at least dozens of runs with different initial points to find more improved solutions. Since our simulations set the total number of runs for LEACH-CKM to 30, the overall computational complexity of LEACH-CKM increases significantly. At each iteration of message-passing, LEACH-AP requires N^2 operations all over the network, while LEACH-XMP requires $N^2 \log N$ operations since LEACH-XMP involves the sorting operation in α_{ij} message calculation, which is slight increase in complexity. Note that LEACH-XMP efficiently finds a near optimal solution without multiple runs of different initialization setups and rapidly converges within dozens of iterations as will be confirmed in the following simulations results. Overall, LEACH-XMP has comparable complexity over other LEACH variations.

Table 2. Computational Complexity.

Protocol	Complexity
LEACH-C	$O(tKN)$
LEACH-CE	$O(tKN)$
LEACH-CKM	$O(tKN)$
LEACH-AP	$O(tN^2)$
LEACH-XMP	$O(tN^2 \log N)$

Figure 6 shows the impact of the BS position and energy distribution. Five simulation scenarios A-E are considered as listed in Table 3. Five columns of each case correspond to LEACH-C, LEACH-CE, LEACH-CKM, LEACH-AP, and LEACH-XMP, respectively. Three stacked bars in each column are associated with the times elapsed until 50%, 90%, and 99% of the nodes are out of operation,

respectively. In all cases, LEACH-XMP outperforms in terms of the lifetime of network, irrespective of the position of the BS. In particular, the 50%-node-survival time is enhanced considerably for the proposed algorithm. This shows that LEACH-XMP sophisticatedly optimizes nonlinear energy consumption which has not been properly handled previously.

Table 3. Simulation Scenarios.

Case	Scenario Description
A	BS = (50, 175), $\mathcal{E}_0$ = 2 J
B	BS = (50, 125), $\mathcal{E}_0$ = 2 J
C	BS = (50, 150), $\mathcal{E}_0$ = 2 J
D	BS = (50, 200), $\mathcal{E}_0$ = 2 J
E	BS = (50, 175), $\mathcal{E}_0$ = unif(1,3) J
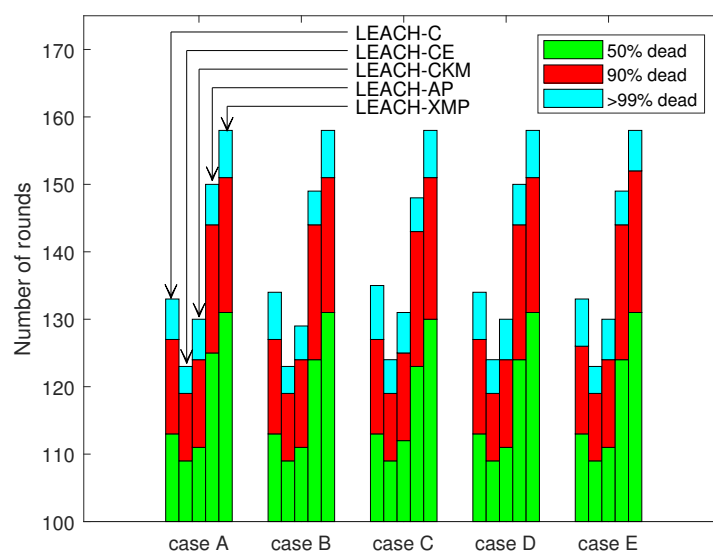
Figure 6. Comparison of network lifetime for various configurations.

Figure 7 shows how the total energy consumption of the network on a single round varies with the number of sensor nodes. As the network grows, the total energy consumption increases gradually. As compared to existing LEACH variations, message-passing-based techniques, i.e., LEACH-AP and LEACH-XMP, provide energy-efficient solutions. Furthermore, the performance improvement of the proposed LEACH-XMP over LEACH-AP becomes noticeable as the network size increases, in that the proposed LEACH-XMP finds a more energy-efficient solution than LEACH-AP from the consideration of the generalized energy consumption model for CH nodes.

Figure 8 shows how the average number of clusters changes depending on network size. As the network grows, the average number of clusters increases in both LEACH-AP and LEACH-XMP. However, the proposed algorithm maintains a small number of clusters comparing to LEACH-AP. LEACH-AP uses a simplified energy consumption model by ignoring non-linear terms. Thus, it normally underestimates the energy consumption of CH nodes, creating more number of clusters. By adopting a realistic energy consumption model, the proposed LEACH-XMP can identify an efficient solution for dynamic cluster sizes in large-scale networks. This explains the total energy consumption performance in Figure 7.
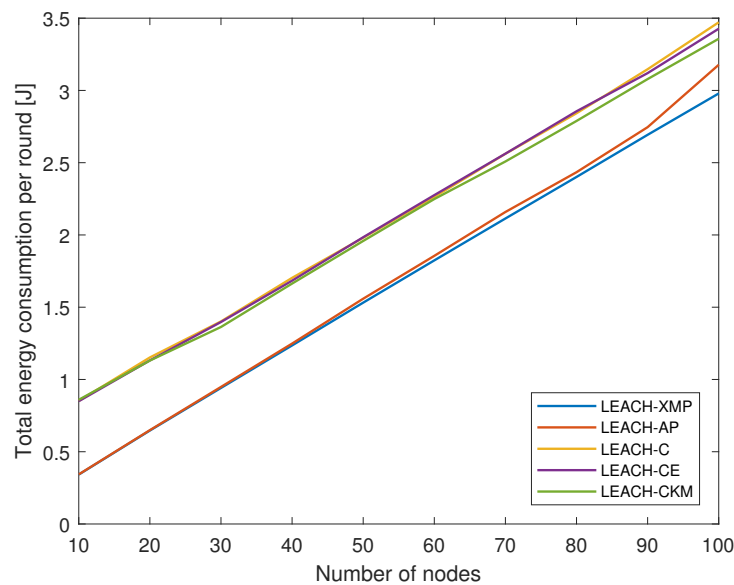
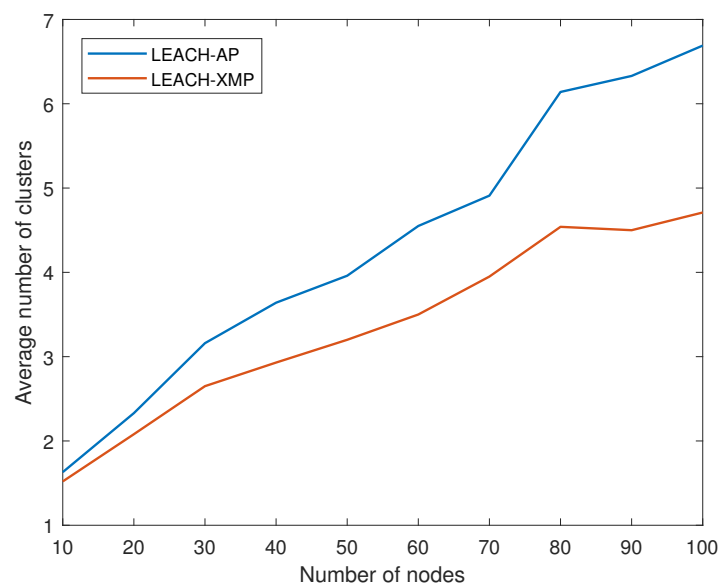**Figure 7.** Total energy consumption per round versus number of sensor nodes.

Figure 8. Number of clusters versus number of sensor nodes.

We now consider a general case of λ, $(\lambda < 1)$, i.e., non-ideal compression at CH nodes. According to (4) and Figure 2, the compressed data size to be delivered from CH nodes to the BS varies depending on λ and cluster size. The compressed data size dominates the energy consumption of CH nodes and the overall cluster formation. Figure 9 shows how the fraction of live nodes reduces as time elapses in the simulation. In all regimes of the λ value, the remaining energy of nodes gradually become depleted and the nodes are excluded from the network. If the cluster formation is identical, a low value of λ leads to the increase in energy consumption of CH as expected from (6). However, the lifetime of the network still extends even in a low λ regime. The clustered results from LEACH-XMP offset the increment of b_n. CH nodes are relieved from the burden to deliver a large amount of the data compressed with a very low rate and saves the remaining energy for survival. Thus, LEACH-XMP outperforms in a general case of $\lambda < 1$ as compared to previously proposed protocols which fail to compensate the effect by lower λ. If $\lambda = 0$, b_n increases quickly as Figure 2 and the cluster formation alone cannot completely compensate for this effect.
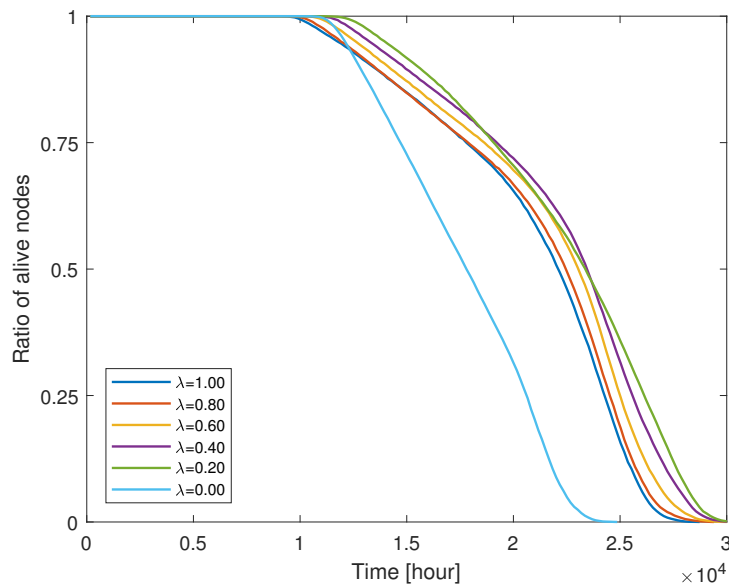
Figure 9. Measurements of live nodes per time for various λ configurations.

Figure 10 illustrates how the average number of clusters varies with the rate of the data compression. The average number of clusters gradually increases with λ. With small λ, the amount of compressed data at CH nodes is large, causing the energy consumption burden to the CH nodes. Accordingly, the network becomes energy-efficient when it maintains a lower number of clusters. By contrast, the amount of the compressed data at CH nodes is small with a large λ. This implies a lower energy consumption burden to the CH nodes, thereby populating more clusters in the network.
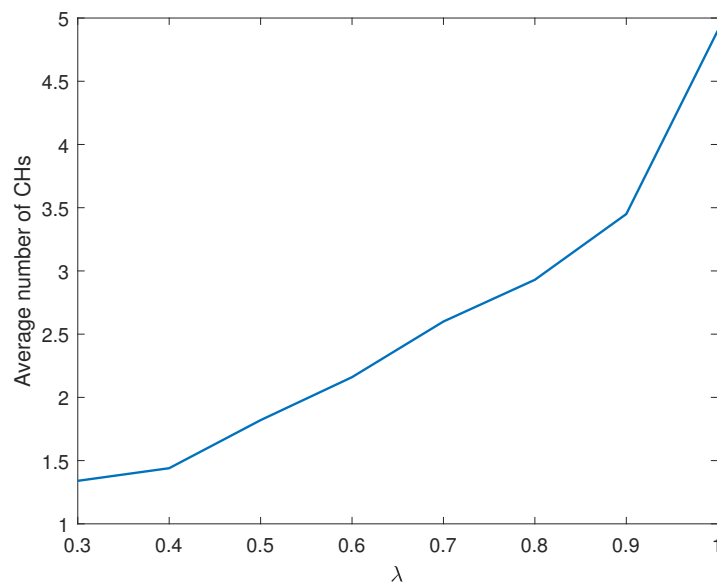

Figure 10. Average number of cluster versus λ.

Figure 11 shows how fast the proposed LEACH-XMP converges in various network sizes. In all ranges of the network size, the proposed LEACH-XMP converges at the latest within 35 iterations. In general, a larger network size requires slightly longer convergence time. However, the performance difference becomes indistinguishable after 10 to 20 iterations. With the low signaling overhead, the proposed LEACH-XMP is beneficial for practical implementations.
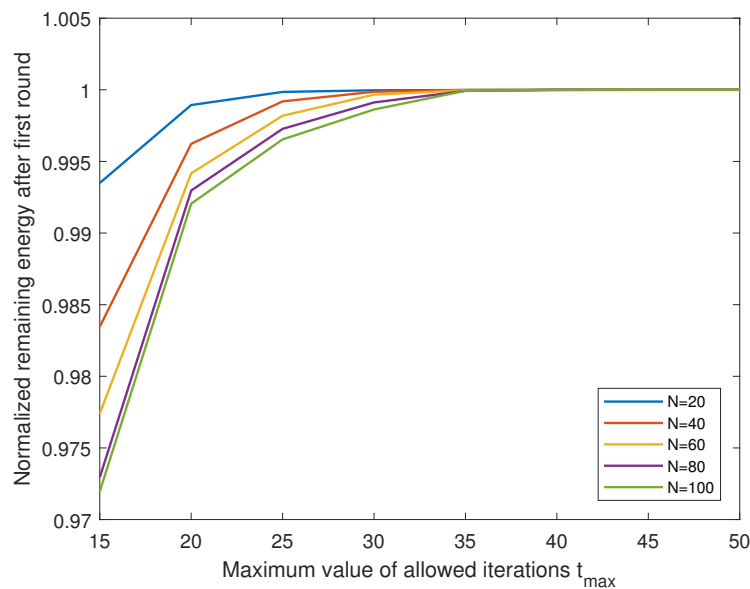
Figure 11. Convergence of LEACH-eXtended Message-Passing (LEACH-XMP).

5. Conclusions

This work develops LEACH-XMP, a LEACH improvement via an enhanced message-passing framework, which maximizes the remaining energy and thus the lifetime of the WSN by considering realistic LEACH-based model. To suggest an efficient solution for this model, nonlinear energy consumption for CH nodes is included in the network energy consumption minimization, enabling an advanced energy-efficient cluster construction. By utilizing the proposed protocol, individual nodes can adaptively form an energy-efficient cluster network regardless of the compression rate of the data. The proposed LEACH-XMP consistently outperforms existing LEACH techniques without absolute positioning of each sensor, predetermination of the number of clusters, and centralized control of the BS. Such an advantage of LEACH-XMP originates from the fact that LEACH-XMP sophisticatedly optimizes nonlinear realistic energy consumption accommodating energy-efficient compressed representation of the sensing data, which has not been properly handled in existing LEACH techniques.

Author Contributions: Conceptualization, I.S. and S.H.L.; Methodology, S.H.L.; Software, J.K.; Validation, J.K.; Formal Analysis, I.S. and J.K.; Writing—Original Draft Preparation, S.H.L.; Writing—Review & Editing, J.K. and I.S.; Funding Acquisition, I.S. and S.H.L.

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

Abbreviations

The following abbreviations are used in this manuscript:

LEACH	Low-Energy Adaptive Clustering Hierarchy
WSN	Wireless Sensor Network
PEGASIS	Power-Efficient Gathering in Sensor Information Systems
HEED	Hybrid Energy-Efficient Distributed clustering
EEUC	Energy-Efficient Uneven Clustering
LEACH-XMP	LEACH-eXtended Message-Passing

LEACH-AP	LEACH-Affinity Propagation
LEACH-C	LEACH-Centralized
LEACH-CE	LEACH-Centralized Efficient
LEACH-CKM	LEACH-C with K-Means and Minimum Transmission Energy routing protocol
CH	Cluster Head
BS	Base Station

References

1. Chong, C.Y.; Kumar, S.P. Sensor networks: Evolution, opportunities, and challenges. *Proc. IEEE* **2003**, *91*, 1247–1256. [CrossRef]

2. Díaz, V.H.; Martínez, J.F.; Martínez, N.L.; del Toro, R.M. Self-Adaptive Strategy Based on Fuzzy Control Systems for Improving Performance in Wireless Sensors Networks. *Sensors* **2015**, *15*, 24125–24142. [CrossRef] [PubMed]

3. Long, H.; Liu, Y.; Wang, Y.; Dick, R.P.; Yang, H. Battery allocation for wireless sensor network lifetime maximization under cost constraints. In Proceedings of the 2009 IEEE/ACM International Conference on Computer-Aided Design, San Jose, CA, USA, 2–5 November 2009; pp. 705–712. [CrossRef]

4. Cheng, C.T.; Tse, C.K.; Lau, F.C.M. An Energy-Aware Scheduling Scheme for Wireless Sensor Networks. *IEEE Trans. Veh. Technol.* **2010**, *59*, 3427–3444. [CrossRef]

5. Yetgin, H.; Cheung, K.T.K.; El-Hajjar, M.; Hanzo, L.H. A Survey of Network Lifetime Maximization Techniques in Wireless Sensor Networks. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 828–854. [CrossRef]

6. Heinzelman, W.B.; Chandrakasan, A.P.; Balakrishnan, H. An application-specific protocol architecture for wireless microsensor networks. *IEEE Trans. Wirel. Commun.* **2002**, *1*, 660–670. [CrossRef]

7. Lindsey, S.; Raghavendra, C.S. PEGASIS: Power-efficient gathering in sensor information systems. In Proceedings of the IEEE Aerospace Conference, Big Sky, MT, USA, 9–16 March 2002; Volume 3, pp. 1125–1130. [CrossRef]

8. Younis, O.; Fahmy, S. HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE Trans. Mob. Comput.* **2004**, *3*, 366–379. [CrossRef]

9. Li, C.; Ye, M.; Chen, G.; Wu, J. An energy-efficient unequal clustering mechanism for wireless sensor networks. In Proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems, Washington, DC, USA, 7–10 November 2005; pp. 598–604. [CrossRef]

10. Tripathi, M.; Battula, R.B.; Gaur, M.S.; Laxmi, V. Energy Efficient Clustered Routing for Wireless Sensor Network. In Proceedings of the 2013 IEEE 9th International Conference on Mobile Ad-hoc and Sensor Networks, Dalian, China, 11–13 December 2013; pp. 330–335. [CrossRef]

11. Mechta, D.; Harous, S.; Alem, I.; Khebbab, D. LEACH-CKM: Low Energy Adaptive Clustering Hierarchy protocol with K-means and MTE. In Proceedings of the 2014 10th International Conference on Innovations in Information Technology (IIT), Al Ain, UAE, 9–11 November 2014; pp. 99–103. [CrossRef]

12. Sohn, I.; Lee, J.H.; Lee, S.H. Low-Energy Adaptive Clustering Hierarchy Using Affinity Propagation for Wireless Sensor Networks. *IEEE Commun. Lett.* **2016**, *20*, 558–561. [CrossRef]

13. Frey, B.J.; Dueck, D. Clustering by Passing Messages Between Data Points. *Science* **2007**, *315*, 972–976. [CrossRef] [PubMed]

14. Kschischang, F.R.; Frey, B.J.; Loeliger, H.A. Factor graphs and the sum-product algorithm. *IEEE Trans. Inf. Theory* **2001**, *47*, 498–519. [CrossRef]

15. Schenato, L.; Fiorentin, F. Average TimeSynch: A consensus-based protocol for clock synchronization in wireless sensor networks. *Automatica* **2011**, *47*, 1878–1886. [CrossRef]