Abstract—In this paper, we propose a cross-layer optimization scheme named Adjusting the Transmission Radius (ATR), which is based on the Energy Consumed uniformly Connected K-Neighborhood (EC-CKN) sleep scheduling algorithm in wireless sensor networks (WSNs). In particular, we discovered two important problems, namely, the death acceleration problem and the network isolation problem, in EC-CKN-based WSNs. Furthermore, we solve these two problems in ATR, which creates sleeping opportunities for the nodes that cannot get a chance to sleep in the EC-CKN algorithm. Simulation and experimental results show that the network lifetime of ATR-Connected-K-Neighborhood-based WSNs increases by 19%, on average, and the maximum increment is 41%. In addition, four important insights were discovered through this research work and presented in this paper.

Index Terms—Death acceleration, duty-cycled, industrial wireless sensor networks (IWSNs), network isolation.

I. INTRODUCTION

A. Group-Based IWSNs

As we had briefly presented in [1], along with the fast development of various emerging technologies in wireless communication and hardware design, small-size sensor nodes are widely deployed and applied in many large-scale industrial factories for a number of monitoring and controlling applications, e.g., large-scale rotational equipment monitoring and fault diagnosis, large-scale pipeline monitoring, toxic gas leak detection and leak point localization, and concentration monitoring of CO2. Having wireless communication capability, these sensor nodes can organize themselves into networks as industrial wireless sensor networks (IWSNs) [2], which are usually powered by batteries with limited energy. Comparing with traditionally applied field bus in large-scale industrial factories, IWSNs have many advantages, e.g., being cost effective, easy to deploy, and easy to move. In particular, by applying IWSNs, it is possible to avoid using a large number of power lines, which may cause a huge mass on power-line deployment. Although IWSNs have a lot of advantages, compared with traditional field bus, a nature shortage still exists as the used batteries are difficult to be recharged with power lines or impossible to be replaced, particularly within certain applications, e.g., vibration sensor nodes deployed inside a rotating machine or temperature sensor nodes deployed inside a heat exchange tube. Furthermore, it is also impossible to apply wireless charging technologies for these sensor nodes to recharge, since strong interference exists in these kinds of large-scale industrial factories, and the wireless charging distance is still not long enough [3]–[6]. Therefore, studying and proposing efficient energy management schemes is a crucial problem in providing theoretical understanding and core algorithms for IWSNs, particularly when deploying real applications.

Energy consumption in sensor networks is normally affected by many factors, e.g., application required sampling rate, deployment style and density of sensor nodes, hierarchical organization, topology of broadcasting tree, in-network data aggregation and fusion, data transmission schemes, sleep scheduling control, and environmental interference. Among all of these factors, the deployment style and sleep scheduling control are two fundamental factors for IWSNs. For example, the IWSNs deployed in a large-scale petrochemical plant will be expected to conduct multiple monitoring tasks, e.g., densely deploying a group of sensor nodes on a certain equipment for fault monitoring and diagnosis and sparsely and evenly deploying many sensor nodes for toxic gas monitoring and leak point detection [7], [8]. This kind of deployment results in a group-based network structure in IWSNs [9], as shown in Fig. 1.

B. Sleep Scheduling and Network Isolation Problem

Sleep scheduling control is considered as a fundamental factor, since many other factors can be further investigated only when the global network connectivity can be guaranteed via appropriated sleep schedule management. Moreover, adopting the
sleep scheduling scheme is considered as a common approach for saving energy, since it allows sensor nodes to sleep [10].

Network isolation is a situation that may happen when the energy of certain sensor nodes is consumed much faster than its neighbors and eventually cause the breaking of global network connectivity. Although having sleep scheduling control in wireless sensor networks (WSNs) can slow down the energy consumption in terms of the entire network, some critical sensor nodes probably still cannot get enough opportunities to sleep, which will eventually cause the network isolation problem. This problem will happen in the WSNs no matter how the sensor nodes were deployed and organized as long as the sensor nodes are not rechargeable. Since this problem is not avoidable, the key point for releasing it in WSNs is to postpone its appearance as late as possible.

C. Motivation and Contribution

In this paper, we study the network isolation problem in a well-known sleep scheduling algorithm—Connected K-Neighborhood (CKN)-based IWSNs. The (CKN) algorithm is a distributed sleep scheduling algorithm [11], which can reduce the number of active nodes efficiently. It keeps the network k connected, and the number of asleep nodes in the sensor network could be decreased by increasing the value of k in CKN. The algorithm Energy Consumed uniformly Connected k-Neighborhood (EC-CKN) [12], which takes the nodes’ residual energy information as a parameter to decide on whether a node is active or asleep, not only can solve the k-connected neighborhood problem but also can assure that the k-awake neighbor nodes have more residual energy than other neighbor nodes in the current epoch. Although the EC-CKN algorithm performs well, the following questions were not addressed yet in [12].

1) As shown in Fig. 2, in EC-CKN and other previous CKN-based studies [13]–[15], all the nodes adopt the same k value, and some nodes may never get a chance to sleep since they do not have enough neighbor nodes as many as k, which are considered as critical nodes or trouble nodes in this paper. Then, the first question is: How to create opportunities for these critical nodes to sleep?

2) Since these critical nodes may run out of energy very soon, which can cause a series of problems, i.e., the network isolation problem, the connectivity problem, or energy imbalance, then the second question is: How to release these problems to further prolong the network lifetime?

Motivated by the aforementioned two major issues, we propose a cross-layer approach that is based on EC-CKN, which is named Adjusting the Transmission Radius Connected K-Neighborhood (ATR-CKN). As shown in Fig. 3, we break down the research problem into two layers, i.e., physical network layer and medium access control (MAC) layer. The advantage of the ATR-CKN sleep scheduling scheme over the original CKN-family-based sleep scheduling algorithms [11]–[15] is that it can locally adjust a sensor node’s transmission radius in the physical network layer, which allows the node to get a chance to sleep based on the CKN family algorithms’ execution result in the MAC layer. The ATR-CKN algorithm inherits all the major properties of the EC-CKN algorithm. Meanwhile, it also makes a significant new contribution of extending the network lifetime by raising the sleeping rate.
D. Insights

Insights of this research work fall into the following five aspects:

1) ATR-CKN creates sleeping opportunities for critical nodes that have to stay in awake status in EC-CKN.
2) ATR-CKN gains a higher sleeping rate than the CKN-family-based sleep scheduling algorithms, e.g., EC-CKN.
3) ATR-CKN helps keep the network coverage ratio caused by the critical nodes’ quick death.
4) ATR-CKN balances the network energy, which further prolongs the network lifetime.
5) ATR-CKN releases the death acceleration problem of the neighbor nodes caused by the always-awake nodes in EC-CKN.
6) ATR-CKN can help release the network isolation problem in EC-CKN.

The rest of this paper is organized as follows. Section II gives a brief introduction about related work of sleep scheduling schemes. Section III defines the network model, the sleep scheduling model, and some notations. In Section IV, the ATR-CKN sleep scheduling scheme is described with detailed information. Specifics of simulation experiments are presented in Section V, which validate the correctness of the proposed ATR-CKN scheme. Finally, Section VI concludes this paper.

II. RELATED WORK

In the study of WSNs, network lifetime has been defined in various ways [16]–[20], and various mechanisms have been proposed to prolong network lifetime with a good coverage level. One common approach to minimize the energy consumption and extend the network lifetime is to put some sensors in the sleep state and put others in the active state for the sensing and communication tasks. When a sensor is asleep, its processor is turned off, but a timer or some other triggering mechanism may be running to wake up the sensor. Therefore, the energy consumed in the sleep state is only a tiny fraction of that consumed in the active state. A sleep scheduling mechanism allows each sensor to determine when it should switch its state and what state it will switch to.

A. Sleep Scheduling in Hierarchical Networks

In a hierarchical network such as a cluster-based network, sensor nodes organize themselves into clusters, and each cluster has a cluster head. Each cluster head manages the sensor nodes in its own cluster, for communication between the cluster and the base station. Communication between cluster heads and the base station may be multihop through other cluster heads.

In [21], Heinzelman et al. have proposed a distributed algorithm for WSNs (LEACH), in which sensor nodes randomly select themselves as cluster heads with some probability and broadcast their decisions. The remaining sensors join the cluster of the cluster head that requires minimum communication energy. LEACH may be unstable during the setup phase depending on the density of sensors. Qin et al. in [22] proposed a $k$-connected overlapping clustering approach with energy awareness, namely, $k$-OCHE. The basic idea of $k$-OCHE is to select a cluster head by energy availability status, and it can balance energy distributions well, consequently prolonging the network lifetime, and gain a quicker routing recovery time. He et al. [18] designed an energy-efficient surveillance system, which classifies sensors into sentries and nonsentries in each cycle and saves energy by putting the nonsentry nodes to sleep most of the time. Its drawback is that the clocks of the nonsentry sensors may drift in the course of time, and as a result, a sentry may need to transmit an awake beacon repeatedly to wake up the nonsentries. The balanced-energy sleep scheme [23] tries to determine the sleep probability such that the maximal number of sensor nodes would consume energy at the same rate independent of the distance to the base station.

B. Sleep Scheduling in Nonhierarchical Networks

Kumar et al. adopted the randomized independent scheduling mechanism extending network lifetime while achieving asymptotic $K$-coverage [24]. At the beginning of an epoch, each sensor node independently decides whether to become active with probability $p$ or go to sleep with probability $1 - p$. Thus, the network lifetime is increased by a factor close to $1/p$. Tian and Georganas proposed a distributed scheduling mechanism to save energy while preserving sensing coverage [25]. To avoid reducing sensing coverage, this mechanism allows a sensor node to turn off only if its sensing area is completely covered by its neighbors’ sensing areas. Note that this mechanism only considers those neighbors within a node’s sensing area to be potential; hence, a sponsored sector may underestimate the number of sensor nodes that can be turned off. In the coverage-aware sleep scheduling scheme [26], each sensor computes the total overlap area between itself and its active neighbors during each scheduling cycle. It then goes to sleep in the next cycle with a probability proportional to the size of the overlap area. Berman et al. presented a centralized and a distributed algorithm to maximize network lifetime while achieving $K$-coverage [27]. The distributed sleep scheduling and range adjustment [28] periodically determines the set of active nodes and sleep nodes for each cycle. An optimization problem is formulated for a scalable hybrid MAC protocol [29] to concurrently access the channel in a machine-to-machine network.
The drawbacks of these methods are as follows: 1) The computation may be too time consuming to fit into the small computational power of a sensor node when there are a large number of neighbors; and 2) the production cost of the node as well as the node’s overhead is increased.

C. CKN-Based Sleep Scheduling Algorithms

Chang and Tassiulas proposed a (CKN) sleep scheduling scheme [10] to generate a favorable duty-cycled WSNs for geographic routing, and the focus of CKN is to allow only a portion of sensor nodes to be awake to save energy consumption while the global network is still connected by those awake nodes. There are nine novel types of CKN-based sleep scheduling schemes designed by our research group recently: geographic-routing-oriented sleep scheduling GSS [13], geographic-distance-based CKN (GCKN) [14], secured energy-aware CKN (SECKN) [15], energy-consumption-based CKN (EC-CKN) [12], [30], data-contain-oriented sleep scheduling DSS [31], geographic-distance-based connected k-neighborhood for first path GCKNF and geographic-distance-based connected k-neighborhood for all paths GCKNA [32], collaborative-location-based sleep scheduling scheme CLSS [33], and priority-based sleep scheduling PSS [34].

GSS aims to shorten the length of the first transmission path explored by TPGF in a CKN sleep-scheduled WSN, when there is a mobile sink. Two GCKN sleep scheduling algorithms called geographic-distance-based connected-k-neighborhood for first path (GCKNF) and geographic-distance-based connected-k neighborhood for all paths (GCKNA) [32] have been introduced in duty-cycle-based WSNs, which can incorporate the advantage of sleep scheduling and mobility. The GCKNF sleep scheduling algorithm minimizes the length of the first transmission path explored by geographic routing in duty-cycled mobile WSNs. Another scheme, which is called the GCKNA algorithm, reduces on the length of all paths searched by geographic routing in mobile WSNs.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Main Focus</th>
<th>Geographic Routing Requirement Considered</th>
<th>Sink Node Mobility Considered</th>
<th>Normal Sensor Node Mobility Considered</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSS</td>
<td>Data transmission path</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>GCKN</td>
<td>Data transmission path</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>SECKN</td>
<td>Sensory security</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>EC-CKN</td>
<td>Sensor energy</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>DSS</td>
<td>Data content</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>GCKNF</td>
<td>Data transmission path</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>GCKNA</td>
<td>Data transmission path</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>CLSS</td>
<td>Integrate mobile cloud computing</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>PSS</td>
<td>Integrate mobile cloud computing</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

However, all of the CKN-based sleep scheduling schemes overlook one significant fact that some nodes have to stay awake when they do not have enough neighbor nodes as many as k. These nodes will run out of energy very soon to cause a series of problems, e.g., death acceleration and network isolation. Consequently, the sleeping rate will not further raise when the value of k is set as 1 for different node density values in the WSN.

III. NETWORK MODEL

A. Communication Network Model

We consider a multihop WSN where all nodes are alive. We assume that each node has a unique ID. All sensor nodes transmit at the same power level and, hence, have the same transmission range $T_r$. Each sensor node is aware of its geographic location and its one-hop neighbor nodes’ geographic locations. The locations of sensor nodes can be obtained by a Global Positioning System or other localization methods. We assume that sensor nodes can know the location of the base station by receiving the packet from the base station. We adopt the transmission rate as shown in [17]–[19].
All communication and packet transmission are over a single shared wireless channel. A wireless link can be established between a pair of nodes only if they are within the wireless communication range of each other. Our proposed scheme only considers bidirectional links. It is assumed that the MAC layer will mask unidirectional links and pass bidirectional links. We refer to any two nodes that have a wireless link as one-hop or immediate neighbors. Nodes can identify one-hop neighbors by using beacons.

B. Energy Consumption Model

We use the same radio model defined in [19]. The amount of energy required to transmit an \( L \)-bit message over a distance \( x \) is \( E_{TX}(L, x) \) given by

\[
E_{TX}(L, x) = \begin{cases} 
E_{\text{elec}} \cdot L + \epsilon_{fs} \cdot L \cdot x^2, & \text{if } x \leq d_0 \\
E_{\text{elec}} \cdot L + \epsilon_{mp} \cdot L \cdot x^4, & \text{if } x > d_0.
\end{cases}
\]  

(1)

\( E_{\text{elec}} \) is the energy dissipating to power the transmitter or receiver circuitry. The parameters \( \epsilon_{fs} \) and \( \epsilon_{mp} \) are the amount of energy dissipating per bit in the radio-frequency amplifier according to distance \( d_0 \), which is given by

\[
d_0 = \sqrt{\frac{\epsilon_{fs}}{\epsilon_{mp}}}. 
\]  

(2)

The energy consumed by receiving this packet is \( E_{TX}(L, x) \), as shown by

\[
E_{TX}(L, x) = L \cdot E_{\text{elec}}. 
\]  

(3)

C. Notations

1) Network Size \((n)\): The network size is the number of nodes in the network. Sensor nodes are randomly deployed in a square area with side length \( l \).

2) Network Lifetime \((\text{round})\): The network lifetime is defined as the average time that sensor nodes in the network run out of its energy from the beginning, and it is in rounds.

3) Minimum Number of Awake Neighbors in an Epoch for Each Node \((k)\): Through varying the value of \( k \), we can keep the network \( k \)-connected and optimize the geographic routing performance.

4) Transmission Radius \((TR)\): The transmission radius is the effective communication range between the node and another node.

5) Default Transmission Radius \((\text{Default}T_R)\): The default transmission radius is the default communication range of a node in ATR-CKN.

6) Maximum Transmission Radius \((\text{Max}T_R)\): The maximum transmission radius is the maximum communication range of a node in ATR-CKN.

7) Increase Transmission Radius \((\text{Increase}T_R)\): This is the increment of effective transmission radius in every adjustment of ATR-CKN.

IV. PROBLEM STATEMENT

When deploying real WSNs for practical applications, it is extremely important to have a good sleep scheduling algorithm to balance the sensor nodes’ energy consumption. As previously mentioned, EC-CKN is a sleep scheduling algorithm designed for duty-cycled WSNs based on the CKN sleep scheduling scheme proposed by Chang and Tassiulas [10]. EC-CKN inherited the key feature of CKN: “allow only a portion of sensor nodes to be awake to save energy while the global network is still connected by those awake nodes” and further focused on prolonging the network lifetime: “the residual energy owned by the awake nodes determined by the sleep scheduling scheme should be more than that of the asleep nodes determined by the algorithm.”

From the discussion on CKN related schemes in Section II-C, it is observed that EC-CKN is that which is focused on balancing energy consumption and network lifetime, which falls into one of the major problems of this paper: death acceleration problem. Therefore, EC-CKN is selected for comparison, but not other CKN related algorithms. However, in EC-CKN, if a node cannot satisfy the condition of \( k \)-neighborhood, it will stay awake all the time. It causes a problem in EC-CKN and even in other CKN-based sleep scheduling algorithms: There may exist nodes that always need to be awake until they run out of energy. Thus, the low-energy reserved awake nodes will run out of energy very soon. After detailed analysis, our research interests fall into the following three aspects.

A. Speed Up Neighbor Node’s Death

In CKN-based sleep scheduling algorithms, a node has to keep awake if it dissatisfies the condition of \( k \)-neighborhood. As we all know, awake nodes have to consume a lot of energy for sensing and communication actions. If a node stays awake, it will die out much earlier than other nodes, which can get a chance to sleep. When the node runs out of energy, its one-hop neighbor nodes may also dissatisfy the condition of \( k \)-neighborhood. In other words, it speeds up its neighbor nodes’ death.

We analyze the problem in the EC-CKN resultant network, as shown in Fig. 4. Nodes in the four subgraphs of Fig. 4 have the same initial energy. Node B has six one-hop neighbor nodes, as the requirement of the minimum \( k \) value is 2, node B can go to sleep in principle. However, to ensure network connectivity, based on the algorithm’s rules, node B is the only neighbor of node A, and it has to stay awake, as shown in Fig. 4(a). Then, node B runs out of energy in round 26, as shown in Fig. 4(b), in which node A is still alive. However, if nodes A and B can do some adjustments, e.g., let node A get a chance to sleep, as shown in Fig. 4(c), node B can work until round 41, as shown in Fig. 4(d). Thus, node A is a trouble node, which will speed up the death of node B and its other one-hop neighbor nodes.

B. Appear the Network Isolation Problem

The most important problem is that CKN-based sleep scheduling networks will appear as the network isolation problem. As previously discussed, some always-awake nodes die out much earlier than other nodes; this will cause a problem in the network’s connectivity.
As shown in Fig. 5, in EC-CKN-based WSNs, nodes A and B in Fig. 5(a) dissatisfy the condition of 3-neighborhood \((K = 3)\), such that they cannot get any opportunity to sleep and save energy. In Fig. 5(b), after executing the EC-CKN algorithm for 25 rounds, nodes A, B, a, b run out of energy, which leads to two isolated network islands. All nodes run out of energy in the round of 46, as shown in Fig. 5(c), which means that these two network islands have been disconnected for 21 rounds. However, our proposed method can avoid the network isolation problem and keep network connected even after 40 rounds, as shown in Fig. 5(d). It is important to note that we do not consider interference and noise that cause the network isolation problem in IWSNs, which, by itself, will be a worthwhile contribution; this is a potential future work.

C. Reduce the Network Coverage Rate

As sensor nodes are randomly deployed in an area, node density cannot stay at the same level. Nodes that dissatisfy the condition of \(k\)-neighborhood are usually distributed in a sparse area of the network.

In Fig. 6, nodes A and B both dissatisfy the condition of 2-neighborhood \((k = 2)\), and node A is the trouble node, as previously discussed. Node B is deployed in a high-density area, and node A is distributed in a relatively sparse area. If node B runs out of energy, its neighbor nodes will replace timely, whereas the network coverage rate will be much reduced if node A dies out.

V. PROPOSED METHOD

Based on the given analysis, we propose the ATR-CKN, which can be applied in any sleep scheduling algorithm based on CKN. In the CKN-based network, the value of \(k\) is set at the beginning of each epoch, then there may exist nodes that never have neighbors as many as \(k\), and ATR-CKN will create opportunities for these nodes to go to sleep. Before executing any CKN-based sleep scheduling algorithm, each node executes ATR-CKN. We assume that all the nodes’ default transmission radius (\(TR\)) is \(\text{Default}TR\), and the maximum \(TR\) supported by a sensor node’s hardware is \(\text{Max}TR\). Each time, the \(TR\) can be increased by \(\text{Increase}TR\) meters.
First, at the beginning of an epoch, ATR-CKN checks the necessity to execute ATR-CKN, to see whether each node has enough neighbor nodes. If the node does not satisfy the condition of $k$-neighborhood, ATR-CKN will further make sure if the node’s transmission radius has already been set at the maximum value or satisfies $k$-neighborhood, then we consider that it is ready for executing the CKN-based sleep scheduling algorithm. Otherwise, ATR-CKN begins to explore the most suitable transmission radius for the node in the fourth part. To ensure the symmetry of the network transmission, its one-hop neighbor nodes should increase their transmission radii to be the same as the current node if the node adjusts its transmission radius. The ATR-CKN’s execution flowchart is shown in Table II.

### VI. PERFORMANCE EVALUATION

#### A. Experimental Setup

To verify the correctness and effectiveness of the proposed ATR-CKN scheme, we conduct a detailed simulation using the WSN simulator NetTopo\(^2\) [36]. In our simulation, the studied WSN has the network size: $800 \times 600$ m\(^2\). The number of deployed sensor nodes is increased from 100 to 800. The value of $k$ is changed from 1 to 10 (each time increased by 1). For each number of deployed sensor nodes, we use 100 different seeds to generate 100 different network deployments. All simulation parameters [13], [37], [38] are listed in Table III. $MaxT_r$ is set to 120 m, and $DefaultT_r$ is set to 60 m. We set the random initial energy for each node between 6 and 10 J. If a node’s transmission radius is changed, the energy consumption will change based on path loss. The network lifetime is defined as the rounds of duty cycle. In this paper, we do not consider interference and noise in the network model.

\(^2\) NetTopo (http://sourceforge.net/projects/nettopo/) is an open-source software for simulating and visualizing WSNs.

#### B. Promotion of Sleeping Rate

In previous studies, the sleeping rate will stay at a stable level under certain node density. As shown in Fig. 7(a) and (b), when the $k$ value is five and larger, under certain network size, the sleep rate approaches zero gradually.

In Fig. 7(c) and (d), ATR-CKN makes a great difference; it promotes the sleeping rate further to prolong network lifetime. With the appropriate adjustment of the always-awake nodes’ transmission radius, these nodes can get a chance to sleep. The sleeping rate in ATR-CKN doubles that in EC-CKN when their node density is the same.

#### C. Network Lifetime

Foremost, we compare network lifetime. We set the random initial energy for each node between 6 and 10 J. Fig. 8 shows the simulation results of the whole network lifetime by using EC-CKN and ATR-CKN. In Fig. 8, the $x$-axis is the different $k$ values, and the $y$-axis is the value of network lifetime. It can be easily seen that ATR-CKN provides better control of the network sleeping rate through adjusting the value of $k$.

#### D. Average Round of First Death Node

Here, we evaluate EC-CKN and ATR-CKN in terms of the average round of the first nodes running out of energy. In Fig. 9(a), we can observe that the average round of the first dead node in EC-CKN is very low. When the network size is set, regardless of the value of $k$, the average round of the first dead node almost stays at the same level. Hence, a series of problems will occur, e.g., the death acceleration problem, the network isolation problem, and the network coverage problem.
Fig. 7. Comparison of sleeping rates in EC-CKN and ATR-CKN. (a) Sleeping rate in EC-CKN (2-D). (b) Sleeping rate in ATR-CKN (2-D). (c) Sleeping rate in EC-CKN (3-D). (d) Sleeping rate in ATR-CKN (3-D).

Fig. 8. Comparison of network lifetime by using EC-CKN and ATR-CKN. (a) Lifetime of EC-CKN (2-D). (b) Lifetime of ATR-CKN (2-D). (c) Lifetime of EC-CKN (3-D). (d) Lifetime of ATR-CKN (3-D).
Fig. 9. Comparison of average round of the first node running out of energy in EC-CKN and ATR-CKN. (a) Average round of the first node running out of energy in EC-CKN. (b) Average round of the first node running out of energy in ATR-based EC-CKN.

However, as we can see in Fig. 9(b), ATR-CKN is much improved. For example, when $k$ is 1, the average round of the first dead node is doubled in every network size. Moreover, when $k$ is 5 and even larger, the average round of the first dead node will also keep a stable trend, but the stable value is also almost doubled compared with the EC-CKN’s.

In CKN-based sleep scheduling schemes, the lifetime of the first dead node has nothing to do with node density, e.g., EC-CKN, as shown in Fig. 9(a). However, in ATR-CKN, when node density gets larger, the lifetime of the first dead node is much longer. In other words, ATR-CKN delays the death of the always-awake nodes, and it is more apparent when node density gets larger.

E. No-Sleeping-Opportunity Nodes

As previously mentioned, there exist a number of nodes that have to remain in the awake state since it cannot satisfy the condition of $k$-neighborhood. As Fig. 10 shows, the ratio of the always-awake nodes in the network is much higher relative to that in ATR-CKN, regardless of the node density value.

In a sensor network where node density is greater than or equal to $8.3 \times 10^{-4}$ per m$^2$, ATR-CKN can assure the inexistence of always-awake nodes. That is, ATR-CKN gives full play to the $k$ value over the whole network. This is also the reason why ATR-CKN can improve sleeping rate and prolong the lifetime in CKN-based sensor networks.

Fig. 10. Comparison of the ratio of the always-awake nodes in EC-CKN and ATR-CKN. (a) Ratio of always-awake nodes in EC-CKN (2-D). (b) Ratio of always-awake nodes in ATR-CKN (2-D).

VII. CONCLUSION

In this paper, we have proposed a cross-layer optimization scheme called ATR-CKN, which is based on EC-CKN. We compare ATR-CKN, and simulation results show that ATR can prolong the lifetime of WSNs because it creates opportunities to sleep for the nodes that have no opportunity to sleep in EC-CKN. The major contributions and discovered insights of this research work are summarized as the following four aspects.

1) First, ATR-CKN delays the death of always-awake nodes. When the node density is larger, the time when the first dead node appears is later. However, in traditional CKN-based sleeping scheduling algorithms, the lifetime of the first dead node has no direct relationship with the network’s node density.

2) Second, in a sensor network where node density is greater than or equal to $8.3 \times 10^{-4}$ per m$^2$, ATR-CKN can assure the inexistence of always-awake nodes. That is, ATR-CKN gives full play to the $k$ value over the whole network.

3) Third, although increasing the transmission radius of nodes means increasing the energy consumption in nodes, by getting the opportunity to sleep, the overall network lifetime can be still prolonged.

4) Finally, compared with other CKN-based sleeping scheduling algorithms, in ATR-CKN, the $k$ value has a bigger influence on controlling the sensor network, e.g., the sleeping rate.
Appendix

This appendix shows the pseudocode of the connected k-neighborhood (CKN) algorithm in [11].

Pseudocode of connected k-neighborhood (CKN) algorithm
(Run the following at each node $u$)

1. Pick a random rank $\rank_u$.
2. Broadcast $\rank_u$ and receive the ranks of its currently awake neighbors $N_u$. Let $R_u$ be the set of these ranks.
3. Broadcast $R_u$ and receive $R_v$ from each $v \in N_u$.
4. If $|N_u| < k$ or $|N_v| < k$ for any $v \in N_u$, remain awake. Return.
5. Compute $C_u = \{v | v \in N_u$ and $\rank_v < \rank_u \}$. Go to sleep if both the following conditions hold. Remain awake otherwise.
   - Any two nodes in $C_u$ are connected either directly themselves or indirectly through nodes within $u$'s two-hop neighborhood that have rank less than $\rank_u$.
   - Any node in $N_u$ has at least $k$ neighbors from $C_u$.
6. Return.

References


Lei Shu (M’07) received the Ph.D. degree from the National University of Ireland, Galway, Ireland, in 2010.

Until March 2012, he was a Specially Assigned Researcher with the Department of Multimedia Engineering, Graduate School of Information Science and Technology, Osaka University, Suita, Japan. Since October 2012, he has been a Full Professor with Guangdong University of Petrochemical Technology, Maoming, China. Since 2013, he has been a Ph.D. Supervisor with Dalian University of Technology, Dalian, China, and a Master Supervisor with Beijing University of Posts and Telecommunications, Beijing, China. He has also been the Vice Director of the Guangdong Provincial Key Laboratory of Petrochemical Equipment Fault Diagnosis, Guangdong University of Petrochemical Technology. He is the Founder of the Industrial Security and Wireless Sensor Networks Laboratory. He is the author of over 200 papers published in related conference proceedings, journals, and books. His current H-index is 25. His research interests include wireless sensor networks, multimedia communication, middleware, security, and fault diagnosis.

Dr. Shu served as a Cochair for more than 50 various international conferences/workshops, e.g., the IEEE International Wireless Communications and Mobile Computing Conference (IWCMC), the IEEE International Conference on Communications (ICC), the IEEE Symposium on Computers and Communications (ISCC), the IEEE International Conference on Computing, Networking and Communication, and the International Conference on Communications and Networking in China (Chinacom). He also served/will serve as a Symposium Cochair for IWCMC 2012 and ICC 2012, as a General Chair for Chinacom 2014 and the 2015 International Conference on Heterogeneous Networking for Quality, Reliability, Security, and Robustness (Qshine), as a Steering Chair for the 2015 International Conference on Industrial Networks and Intelligent Systems, and as a Technical Program Committee Member of more than 150 conferences, including the IEEE International Conference on Distributed Computing in Sensor Systems, the IEEE International Conference on Mobile Ad hoc and Sensor Systems, ICC, Globecom, IEEE International Conference on Computer Communications and Networks, IEEE Wireless Communications and Networking Conference, and ISCC. He currently serves as an Editor-in-Chief for the European Alliance for Innovation Endorsed Transactions on Industrial Networks and Intelligent Systems and an Associate Editor for a number of renowned international journals. He received the 2010 IEEE Global Communications Conference and 2013 IEEE International Conference on Communications Best Paper Awards. He is a member of the IEEE Communication Society, the European Alliance for Innovation, and the Association for Computing Machinery.

Lei Wang (M’01) received the B.S., M.S., and Ph.D. degrees from Tianjin University, Tianjin, China, in 1995, 1998, and 2001, respectively.

He is currently a Full Professor with the School of Software, Dalian University of Technology, Dalian, China. He was a Member of Technical Staff with Bell Labs Research China during 2001–2004; a Senior Researcher with Samsung, South Korea, during 2004–2006; a Research Scientist with Seoul National University, Seoul, Korea, during 2006–2007; and a Research Associate with Washington State University, Vancouver, WA, USA, during 2007–2008. He has published over 70 papers, which have over 1000 citations. His research interests include wireless ad hoc networks, sensor networks, social networks, and network security.

Dr. Wang is a member of the Association for Computing Machinery and a Senior Member of the China Computer Federation.

Chunsheng Zhu (S’12) received the B.E. degree in network engineering from Dalian University of Technology, Dalian, China, in 2010 and the M.Sc. degree in computer science from St. Francis Xavier University, Canada, in 2012. He is currently working toward the Ph.D. degree in the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC, Canada.

He has around 40 papers published or accepted by refereed international journals, e.g., the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, the IEEE TRANSACTIONS ON COMPUTERS, the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, the IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING, and the IEEE SYSTEMS JOURNAL, and conferences, e.g., the IEEE Global Communications Conference and the IEEE International Conference on Communications. His current research interests include wireless sensor networks and cloud computing.

Jianwei Niu (SM’13) received the Ph.D. degree in computer science from Beihang University, Beijing, China, in 2002.

He is a Professor with the School of Computer Science and Engineering, Beihang University. He has published over 100 referred papers and holds 30 patents in mobile and pervasive computing. His current research interests include mobile and pervasive computing and mobile video analysis.

Dr. Niu has served as an Associate Editor for the International Journal of Ad Hoc and Ubiquitous Computing and the Journal of Network and Computer Applications (Elsevier). He was awarded the Best Paper Award at the 2013 IEEE International Conference on Communications and the 2013 IEEE Wireless Communications and Networking Conference.

Mithun Mukherjee (M’10) received the B.E. degree in electronics and communication engineering from the University Institute of Technology, Burdwan University, Burdwan, India, in 2007; the M.E. degree in information and communication engineering from the Indian Institute of Science and Technology, Shibpur, India, in 2009; and the Ph.D. degree in electrical engineering from the Indian Institute of Technology Patna, Patna, India, in 2015.

He is currently a Postdoctoral Fellow with the Guangdong Provincial Key Laboratory of Petrochemical Equipment Fault Diagnosis, Guangdong University of Petrochemical Technology, Maoming, China. During 2014–2015, he was an Assistant Professor with the Department of Electronics and Communication Engineering, National Institute of Technology Hamirpur, Hamirpur, India. His research interests include wireless sensor networks, sensor network middleware, multimedia communications, device-to-device communications, and dynamic spectrum sharing.