CoDoC: A Novel Attack for Wireless Rechargeable Sensor Networks through Denial of Charge

Chi Lin*†, Zhi Shang*†, Wan Du‡, Jiankang Ren§¶, Lei Wang*†, and Guowei Wu*†
*School of Software Technology, Dalian University of Technology, Dalian 116023, China
†Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, Dalian, 116621, China
‡Department of Computer Science and Engineering, University of California, Merced 95543, U.S.A.
§School of Computer Science and Technology, Dalian University of Technology, Dalian 116023, China
Email: {c.lin, rjk, wgwdut}@dlut.edu.cn, wdu3@ucmerced.edu, lei.wang@ieee.org

***Abstract*—Wireless rechargeable sensor networks (WRSNs), benefiting from recent breakthrough in wireless power transfer (WPT) technology, emerge as very promising for network lifetime extension. Traditional methods focus on scheduling algorithms and system optimization, and the issue of charging security/threat is ignored, causing it vulnerable to attacks. In this paper, we develop a novel attack for WRSN through Denial of Charge (DoC) aiming at maximizing destructiveness. At first, we form a generalized on-demand charging model, which provides fundamental basis for designing charging attacks. Then a request prediction method (RPM) is introduced for predicting the emergences of charging requests. Afterwards, a Collaborative DoC attacking algorithm (CoDoC) is developed, which tempers/modifies and generates fake charging requests, yielding normal nodes exhausted. Finally, to demonstrate the outperformed features of CoDoC, extensive simulations and test-bed experiments are conducted. The results show that, CoDoC outperforms in making sensor exhausted as well as causing missing events.**

I. Introduction

Benefiting from the recent breakthrough in wireless power transfer (WPT) technology, the bottleneck of energy limitation in wireless sensor networks (WSNs) has been eased [1], [2]. Thus, the concept of Wireless Rechargeable Sensor Network (WRSN) has come up and attracts increasingly attentions [3].

In recent years, much effort has been devoted to improving the performance of WRSNs by optimizing the charging scheduling [4]. In general, most research focuses on scheduling algorithms [5], collaborative control [6] and system performance optimizations [7]. However, security issues in WRSN have been overlooked. For example, in the on-demand charging architecture, once a malicious/fake charging request is received by a Wireless Charging Vehicle (WCV), a wrong charging tour may be obtained due to fraud. In that case, a fraction of nodes will be left unserved and exhausted in vain. This kind of Denial of Charge (DoC) attack may destruct the network reliability and functionality and trigger event loss. Even worse, such an attack may lead to catastrophic consequences, especially for real-time [8] and safety-critical applications, such as heart disease monitoring, forest fire alarm, and so on [9], [10]. Therefore, much attention should be paid on attacks and security issues in WRSNs.

Researching the attacking approaches can contribute to designing network threat models, which is quite useful for developing safeguard mechanisms especially for network security area. Therefore, in this work, we focus on attacks in WRSNs and develop a novel attack model for the on-demand architecture, seeking for maximizing the destructiveness by means of creating and tempering unrealistic/fake charging requests.

When developing attacking schemes, we confront with three challenges. As the charging performance directly depends on charging scheduling algorithms, attacking schemes have an one-on-one corresponding relation to the scheduling methods. In practice, it is complicated and useless to customize a large number of unique attacking algorithms, because each of which is dedicated for one scheduling algorithm. As a consequence, the first challenge is: (I) how to propose a generalized on-demand charging architecture model, which is feasible for presenting various kinds of on-demand charging approaches by configuring different parameters? This model paves the way of developing a generalized DoC attack easily, no matter which on-demand charging algorithm is applied. The second problem is: when mounting an attack, the attacker will temper and modify fake charging requests. However, continuously upcoming of sporadic requests cannot baffle the WCV, instead, explosive requests in a short time will cause exhausted nodes resulting from exceeding WCV's serving capabilities. Therefore, the second challenge is: (II) when and how to determine the best time to manipulate malicious nodes to simultaneously send fake requests so as to maximize destructiveness? Besides that, (III) how to disguise the presence of the attack without notice/detection of the WCV is the third challenge.

To tackle above challenges, in this paper, we propose a novel DoC algorithm on WRSN for on-demand charging architectures named CoDoC. At first, we propose a generalized on-demand charging model for developing DoC attack methods. Then a request prediction method (RPM) is introduced for predicting the emergence of charging request. Afterwards, an attack algorithm is developed, through which network destructiveness can be maximized. Finally, to demonstrate the outperformed features of CoDoC, extensive simulations and test-bed experiments are conducted.

In general, the main contributions of this paper are summarized as below:

- To the best of our knowledge, we are the first to concentrate on charging attacks in WRSNs. A Collaborative Denial of Charge (CoDoC) attack, which destructs a

¶The corresponding author: Dr. Jiankang Ren, rjk@dlut.edu.cn.

network by maximizing event loss through intentionally tempering, modifying and generating fake charging requests, is proposed.
- To maximize attacking destructiveness (e.g. maximizing missing point of interests (PoIs) & missing events), we abstract the features of state-of-the-art on-demand charging architecture and develop a generalized model for formalizing the charging scheduling procedure, which provides the fundamental basis for designing charging attacks.
- To find the best opportunity for launching the DoC attack, a request prediction method (RPM) is proposed. RPM can be used to predict the number of upcoming requests initiating in a certain interval, within which charging demand exceeds a WCV's charging capability, yielding some nodes' requests ignored, and exhausted in vain.

The remainder of this paper is organized as follows. Section II gives a brief overview on literature review. Section III introduces background knowledge. In Section IV, a DoC attack algorithm named CoDoC is proposed. Test-bed experiments and simulations are conducted to show the performance in Section V and Section VI. Finally, Section VII concludes this paper and points out the future work.

II. Literature Review

Recently, much effort has been devoted to WRSNs research in aspects of charging security [11], charging scheduling [8], and network optimizations [7]. However, few of them pay attention to the threat models as well as charging attack issues.

In the network security aspect, most research focuses on electromagnetic radiation problem in charging applications [11]. Dai *et al.* [11] proposed the definition of safe charging and maximized the charging utility of charge devices by adjusting the power of mobile chargers. However, these work only care about avoiding the negative effects of electromagnetic radiation on human health. Issues of charging attacks and corresponding countermeasures are never mentioned.

Although existing solutions to security issues in WRSNs are not suitable to refer to, attacks in WSNs [12] (e.g. DoS attack [13], DDoS attack [14], and node capture attack [15]) can still provide insightful guidance for developing destructive and effective charging attacks, which are deserved to be mentioned. Ning *et al.* [13] introduced a special DoS attack, in which the attacker broadcasts fake messages and forces the receiving nodes to waste energy on performing a large number of unnecessary signature verifications. Consequently, sensor nodes will eventually exhaust their battery power. Comparing with the DoS attack, more attacking sources are used in DDoS [14]. In DDoS attack, incoming traffic comes from plenty of IP addresses, hence, countermeasures make little effect by blocking a single address. In the node capture attack [15], Tague *et al.* [15] proposed a metric using circuit mapping to quantify the vulnerability of the traffic, based on which the attacker can always maximize the destructiveness.

As the DoC attack considered in this paper is taken based on the on-demand charging architecture, several state-of-the-art scheduling methods are summarized for reference. He *et al.* [16] proposed the Nearest-Job-Next with Preemption (NJNP) on-demand architecture, which always serves spatial closest nodes. Stankovic *et al.* [17] designed a Earliest-Deadline-First (EDF) for the on-demand architecture, which aims to serve the most emergency nodes. Lin *et al.* [18] proposed a mixed priority based charging scheduling algorithm (mTS), which takes into account temporal and spatial factor synthetically.

However, all research achievements on WRSNs overlook the influences of attacks. In this work, we concentrate on such threats and propose a novel DoC attack to damage network functionality for on-demand charging architecture.

III. On-demand Architecture Generalization

In this section, at first, we present related backgrounds. Then, we generalize the architecture. Finally, we formalize our objective problem.

A. Network Model

Typically, a WRSN is composed of three components: a base station (BS), a wireless charging vehicle, and a number of rechargeable sensor nodes. BS is responsible for collecting, aggregating and mining sensory data from sensors, and it can also provide battery provisioning and replacing service for WCV. WCV is applied to replenish energy for rechargeable sensors and acts as the energy delivering medium throughout the network. It is equipped with a transmitting coil and is able to wirelessly charge the battery of the sensors. The energy capacity of WCV is limited. Once its residual energy is low, it will return to the BS for energy replacement. Homogenous sensors implemented with receiving coils are deployed to monitor point of interests (PoIs) in the area, and they can harvest energy from the WCV when the distance of two coils is within a certain range through WPT. In our model, events (i.e. PoIs) may happen anywhen and anywhere throughout the network area and they are regarded as equally important.

B. Charging Scheduling Scheme

As a classic on-line charging architecture in WRSNs, the on-demand architecture shows merits in flexibility and scalability with dynamic topology. Through collecting charging requests from "hungry" nodes, WCVs will be employed to patrol over the network for replenishing energy. Hence, we focus on designing attacking algorithms on such an architecture. The main reason that we do not choose the off-line scheduling methods such as periodically scheduling [19] is: the charging sequences of nodes are determined in network initialization, which lacks of flexibility. Thus, malicious attacks will fail to make destructions because the scheduling algorithm has not been changed at all.

In the on-demand charging architecture, as charging scheduling schemes (i.e. NJNP [16], EDF [17], and mTS [18]) vary from each other, leading to various charging tours/sequences, it is essential to formalize a general model for them, which will provide the fundamental basis for developing attacking methods. We hereby design a general on-demand charging scheduling model. As both distance and charging deadline are considered in the process of scheduling, the general model should be considered to be highly adaptable.

During the working process of WRSNs, once the remaining energy of sensor n_i falls below a specific threshold θ, a charging request including node ID *i*, node location (x_i, y_i), current energy consumption rate q_i, and current time T_i, expressed as $<i, x_i, y_i, q_i, T_i>$, will be initiated by sensor n_i and delivered to the WCV.

From the point view of WCV, all received charging requests will be recorded in a waiting queue Ψ. Hence, to select an appropriate request to serve, a straightforward method is to designate each node a priority. The node with the highest priority will be served first. Whenever a charging task is completed, the waiting queue will be updated. Now, we will detailedly demonstrate how to designate a priority of a charging request (node).

1) Priority determination: In the on-demand charging architecture, to designate appropriate priorities for requests, we pay close attention to the temporal and spatial characteristics. Here, two different priorities, temporal priority and spatial priority, are calculated accordingly.

The temporal priority $\lambda^{(t)}(i)$ of node n_i is defined as its own remaining lifetime t_i, which relates to the residual energy e_i and energy consumption rate q_i, and it is calculated as $t_i = \frac{e_i}{q_i}$.

Then, similarly, spatial priority $\lambda^{(d)}(i)$ is defined as the distance D_i between node n_i and WCV.

By combining two priorities together, we introduce the definition of a mixed priority $\lambda^{(m)}(i)$. Here, two parameters α and β are defined as weights of temporal and spatial priorities respectively as follow:

$$\lambda^{(m)}(i) = \alpha\lambda^{(t)}(i) + \beta\lambda^{(d)}(i) \ . \tag{1}$$

Note that, for a node n_i, a smaller mixed priority $\lambda^{(m)}(i)$ will lead to a higher charging priority.

2) Reachable-in-Time Test: A straightforward usage of the mixed priority is to select the charging candidate: a node with the smallest mixed priority value $\lambda^{(m)}(i)$ will be designated the highest charging priority by WCV.

However, as both temporal and spatial priorities are involved, in case of setting parameters α and β inappropriately, some extreme case should be focused on, such as: a remote node may be designated a high priority. Thus, to avoid this phenomenon, Reachable-in-Time Test and Residual-Energy Test are introduced. Reachable-in-Time Test aims to check whether the node is reachable before exhausting its energy. Residual-Energy Test is used to identify whether WCV's residual energy is sufficient for returning back to BS after finishing the current assignment.

Reachable-in-Time Test: Once the node with the highest priority is selected, WCV should calculate the time to arrive at the node and compare it with the node's charging deadline. If WCV is not able to reach the node before its deadline, the Reachable-in-Time Test will turn to the node with the second-highest priority and the node with highest priority will be identified as a dead node.

This decision-making process can be formulated as follow:

$$\text{Reachable-In-Time Test} = \begin{cases} \text{serve}, & t_i > \frac{D_i}{V} \\ \text{remove}, & t_i \leq \frac{D_i}{V}. \end{cases} \tag{2}$$

Here, t_i denotes the residual lifetime of n_i and D_i is the distance between WCV and node n_i.

3) Charging time calculation: After passing the Reachable-In-Time Test, we tend to determine how long it takes for charging n_i. The charging time for target n_i can be decided by the energy receiving rate q_r and its residual energy e_i. The rate of energy received by sensor nodes, q_r, can be calculated as:

$$q_r = \varepsilon * q_c \ , \tag{3}$$

where ε denotes the ratio of energy obtained by sensor nodes from WCV. q_c is the energy charging rate of WCV.

Then the residual energy of n_i can be calculated as:

$$e_i = \theta - (T - T_i)q_i \ . \tag{4}$$

Here, e_i and q_i are the residual energy and energy consumption rate of n_i. θ is the energy warning threshold, T and T_i are current time and request sending time of n_i. Since the transmission delay is negligible compared to the service time [19], T_i can also be regarded as the receiving time of request.

After energy receiving rate q_r and residual energy e_i are calculated, the charging time for n_i is computed as:

$$t_i^{(c)} = \frac{C^{(n)} - e_i}{q_r} \ . \tag{5}$$

Here, $t_i^{(c)}$ denotes the charging time of n_i serviced by WCV, $C^{(n)}$ is the battery capacity of nodes.

Residual-Energy Test: After determining the service time for selected node n_i, it is necessary for WCV to confirm whether its residual energy is enough to return back to BS after finishing the charging task. If so, it will move to node n_i; otherwise, it will turn to BS and get energy from it.

$$\text{Residual-Energy Test:} = \begin{cases} n_i, & \frac{E_w - (D_i + D_i^{(BS)}) \cdot q_m}{q_c} > t_i^{(c)} \\ \text{BS}, & \frac{E_w - (D_i + D_i^{(BS)}) \cdot q_m}{q_c} \leq t_i^{(c)} \end{cases} , \tag{6}$$

where E_w refers to the residual energy of WCV, q_c and q_m are the charging and moving energy consumption rates of WCV. $D_i^{(BS)}$ denotes the distance between node n_i and BS.

C. General Model for On-Demand Charging Scheduling

Based on above observations, we formalize the general model of the on-demand scheduling scheme in Algorithm 1.

Algorithm 1 proceeds as follows. Firstly, a set of nodes N are taken as input parameters. Then nodes deliver their charging requests to WCV when they need energy replenishment, and WCV calculates the priorities for them (Line 4-8). Afterwards, WCV sorts these charging requests and checks whether the request with the highest priority is reachable before its charging deadline so as to choose a charging target (Line 9-14). Then, WCV moves towards the target and serves it (Line 15). Finally, the waiting queue will be updated after a charging task has been completed (Line 16).

A prominent feature of our generalized model is that, by means of setting different variables (α, β), our model will be customized into specific charging schemes. For example, when $\alpha = 1$ and $\beta = 0$, the model is specialized as EDF [17], $(0.5, 0.5)$ refers to mTS [18], and NJNP [16] is formalized as $(0, 1)$. This achievement lays the fundamental basis for developing attacking schemes on this generalized model.

Algorithm 1 Generalized model of on-demand charging scheduling

Input: A node set N
Output: Charging target node n_i
Initialize parameters: α, β;
Construct a request queue Ψ;
while $|\Psi| > 0$ **do**
  for $i \leftarrow 1$ to $|\Psi|$ **do**
    Get t_i and $d_{i,j}$;
    Calculate $\lambda^{(t)}(i)$ and $\lambda^{(d)}(i)$;
    Calculate $\lambda^{(m)}(i)$;
  end for
  Sort $\Psi^{(m)}$ by $\lambda^{(m)}(i)$;
  Choose the first node n_i in the sorted queue;
  while n_i cannot pass Reachable-In-Time Test **do**
    Remove n_i from Ψ;
    Re-select the first node in Ψ as n_i;
  end while
  Charge n_i;
  Update the request queue: Ψ.
end while

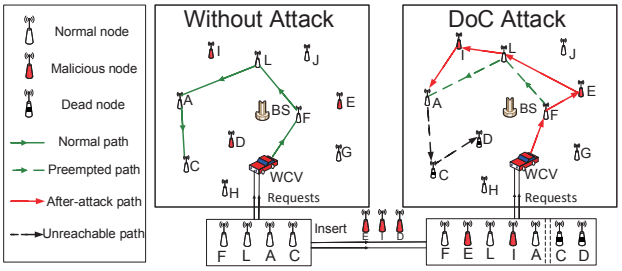


Fig. 1. Node exhaustion caused by DoC attack

Next, we will formalize the adversarial model based on this on-demand charging architecture.

D. Adversary Model

Before introducing the attacking process, we model the capability of an adversary in advance. In this work, an adversary is able to eavesdrop on data flow of sensors [20]. He has the knowledge of the network protocol [21] and knows the speed of WCV.

When launching an attack, an adversary first eavesdrops the network communications to gather sufficient information including node identifications, locations, authenticated neighbors, verification keys, etc. These information will be used to capture the target node [20].

After compromising a node m_i, information such as id, location, and energy consumption rate q_i is considered to be compromised. Hence, the malicious node is able to send a fake charging request to WCV. As id and location information of node m_i are stored in WCV and BS, therefore, the only way to cheat WCV or BS is to modify the request initiating time T_i and energy consumption rate q_i. Upon the receptions of such unexpected malicious and fake requests, the ranking of nodes based on mixed priorities will be consequently changed. Therefore, some nodes which are ranked with a high priority and surely served by WCV may be eventually ignored, and starved to death. We name this novel attack in WRSNs as Denial of Charge (DoC). In our model, the number of nodes captured by an attacker is $|M|$. The nodes are regarded as equally important because the monitored events may happen anywhen and anywhere throughout the network area.

For ease of simplicity, we give an example in Figure 1. As shown in Figure 1, WCV serves sensor nodes under the on-demand charging architecture. Originally, n_C will be served. However, once DoC is launched, several malicious and fake requests will be injected into the network. Thus, more not-so-urgent/important nodes in Ψ will be serviced. Consider two nodes, n_C and n_D, after DoC attack is launched, the charging tour and serving time of the WCV will be prolonged simultaneously. Thus, nodes with low priorities, no matter malicious or not, will be left unserved due to the long lasting waiting time, and exhaust in vain.

1) Problem statement and objective: The aim of an attacker is to destroy the integrity and sensitivity of the network. More exactly, we intend to maximize the destructiveness to the network by launching DoC attack.

Problem Statement: *In a WRSN containing a BS, a series of sensor nodes and a WCV, an attacker has already captured $|M|$ sensor nodes as malicious nodes. How does the attacker manipulate the malicious nodes to collaboratively send fake charging requests so as to change sequence of on-demand charging scheduling for maximizing the destructiveness to the network?*

As mentioned before, in our network, events may happen anytime and anywhere throughout the network, and event missing may cause catastrophic consequences, therefore, to evaluate the destructiveness of the attack, we focus on how many events are missing. Hence, the destructiveness is quantified based on the missing events, and it is denoted by η. Here, we formalize the problem as a single-target optimization problem with the goal as:

$$max \quad \eta = \sum^{i \in P} \mu_i \ , \tag{7}$$

where P is the set of PoIs and μ_i is calculated as:

$$\mu_i = \begin{cases} 1, & \sum^{d_{i,j} \in (0,R]} \varphi_j = 0 \\ 0, & \sum^{d_{i,j} \in (0,R]} \varphi_j > 0 \ . \end{cases} \tag{8}$$

Here, $d_{i,j}$ refers to the distance between PoI p_i and node n_j. R is the coverage range of homogeneous sensor nodes. φ_j is the living state of n_j, which can be defined as:

$$\varphi_j = \begin{cases} 0, & n_j \ is \ dead \\ 1, & n_j \ is \ alive \ . \end{cases} \tag{9}$$

Therefore, a PoI p_i will be missed if and only if all sensor nodes monitoring this point are exhausted.

Usually, malicious nodes may: (1) pass the Reachable-in-Time Test but exhaust its energy when WCV arrives; (2) send a charging request immediately after getting charged from WCV; (3) refuse to send charging requests until drain its own energy; (4) send multiple fake requests at the same moment T or in the same malicious consumption rate $q^{(m)}$. These behaviours can be easily recognized by BS.

Therefore, to ensure that the attacking activity cannot be directly identified by WCV or BS, an external restraining mechanism is necessary to be set up. In order to formulate the constraint that a fake request needs to satisfy, we define $\Theta_i^{(m)}$ as the energy warning threshold designed for malicious node m_i. The DoC attack will not be launched only when the energy of malicious node meets the following constraints:

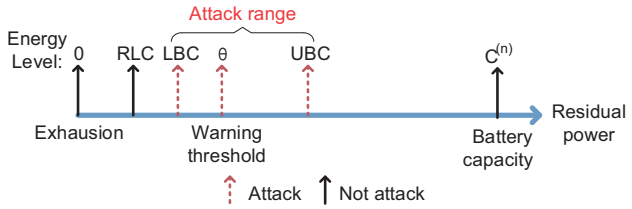


Fig. 2. The constraints on malicious nodes

Rest-Lifetime-Constraint: Suppose that a node m_i becomes the highest-priority node and the Reachable-in-Time Test has been passed. Unfortunately, m_i has already exhausted when WCV reaches there. Thus, the malicious node m_i will be checked out because its actual lifetime is shorter than the fake lifetime, which can be formalized as:

$$\Theta_i^{(m)} \geq \frac{\Theta * q_i}{q_i^{(m)}} \ , \tag{10}$$

where q_i and $q^{(m)}$ denote the actually energy consumption rate and malicious energy consumption rate of m_i, respectively.

Upper-Bound-Constraint: Suppose that if nodes are allowed to send charging requests immoderately, to avoid other sensor nodes receiving charging services, malicious nodes tend to send charging requests immediately after getting charged from WCV. Thus, the charging receiving queue will always be taken up by malicious nodes. In fact, to avoid receiving this kind of malicious requests, WCV sets an Upper-Bound-Constraint for each sensor node as follow:

$$\Theta_i^{(m)} \leq C^{(n)} - (C^{(n)} - \Theta)\frac{q_i}{q_{max}} \ , \tag{11}$$

which denotes that even if the node n_i is working with a possible maximum energy consumption rate q_{max}, it is impossible to send charging request so frequently.

Lower-Bound-Constraint: Similar to the Upper-Bound-Constraint, an Lower-Bound-Constraint is set for each sensor node. As a result, the malicious node, which intends to deliberately exhaust its own energy by refusing to send charging request, will be checked out.

$$\Theta_i^{(m)} \geq max\{C^{(n)} - (C^{(n)} - \Theta)\frac{q_i}{q_{min}}, 0\} \tag{12}$$

Equation (12) indicates that even if the node n_i is working with a possible minimum energy consumption rate q_{min}, it is impossible to send request very infrequently.

Overall, the objective problem described in Equation (7) must subject to the following constraint:

$$\begin{aligned}\Theta_i^{(m)} \in (max\{C^{(n)} - (C^{(n)} - \Theta)\frac{q_i}{q_{min}}, \frac{\Theta * q_i}{q_i^{(m)}}\}, \\ C^{(n)} - (C^{(n)} - \Theta)\frac{q_i}{q_{max}}) \ ,\end{aligned} \tag{13}$$

which is applied to all sensor nodes in the network.

To illustrate the above analysis, constraints on the residual power of malicious nodes are depicted in Figure 2. With the aspiration of sending fake requests, the energy levels of malicious nodes should not only fall below the Upper-Bound-Constraint (UBC) but also over both Rest-Lifetime-Constraint (RLC) and Lower-Bound-Constraint (LBC).

IV. The Proposed Scheme

To launch an effective DoC attack, we first design a request prediction method, which is used to calculate attacking time for all malicious nodes to initiate fake requests. Then, to further maximize the destructiveness of the attack, we develop a collaborative DoC attacking algorithm named CoDoC.

A. Request Prediction Method

As malicious nodes aim to change the charging sequence of the request queue, the number of coming requests during each decision-making cycle must be predicted to maximize destructiveness. However, from the point view of the adversary, the energy consumption rate q_i of each node cannot be obtained. Thus, to predict q_i as well as T_i in a coming request <i, x_i, y_i, q_i, T_i>, a request prediction approximate algorithm is provided.

To formulate the process of data transfer, we follow the same energy consumption model as in [22]:

$$E_T(k,d) = (E_t + \sigma * d^2) * k \ , \tag{14}$$

$$E_R(k) = E_r * k \ . \tag{15}$$

Here, $E_T(k,d)$ and $E_R(k)$ denote the power consumption of data transmission and reception, respectively. d is the distance between transmitter and receiver. k denotes the total length of transmitted data bit. E_t, E_r, σ are constant parameters obtained by communication experiments [22], $E_t = E_r = 558nJ/bit$ and $\sigma = 44.66pJ/bit/m^2$.

To calculate the energy consumption rate q_i of node n_i, we first subdivide the energy consumption rate q_i into energy reception consumption rate $q_i^{(R)}$ and energy transmission consumption rate $q_i^{(T)}$.

Then, we focus on the energy reception rate $q_i^{(R)}$ of n_i, and we define $\overline{I_t(i)}$ as the average number of data bits received by n_i in a time interval t:

$$\overline{I_t(i)} = \frac{(\sum^{j\in\tau_i} | < j,i > |) \cdot \bar{b}}{t} \ . \tag{16}$$

Here, τ_i is the set of neighbors of n_i, indicating that a communication link $l_{j,i}$ is able to be constructed. $\bar{b}$ denotes the average number of transmitted data bits in a data packet. $| < j,i > |$ is the total number of data packets transmitted from n_j to n_i.

Similarly, the average data bits flowing out from n_i in a time interval t, denoted as $\overline{O_t(i)}$, is calculated as:

$$\overline{O_t(i)} = \frac{(\sum^{j\in\tau_i} | < i,j > |) \cdot \bar{b}}{t} \ . \tag{17}$$

By combining Equation (15) and Equation (16), we can get the energy reception consumption rate $q_i^{(R)}$ as:

$$q_i^{(R)} = \frac{E_r \cdot \bar{b}(\sum^{j\in\tau_i} | < j,i > |)}{t} \ . \tag{18}$$

The transmission consumption rate $q_i^{(T)}$ can be calculated referring to Equation (14) and Equation (17) as:

$$q_i^{(T)} = \frac{\bar{b} \cdot [E_t \sum^{j\in\tau_i} | < i,j > | + \sigma(\sum^{j\in\tau_i} | < i,j > | d_{i,j}^2)]}{t} \ . \tag{19}$$

Thus, the energy consumption rate q_i can be calculated by Equation (18) and Equation (19) as:

$$q_i = \frac{\bar{b} \cdot [E_r \cdot (\sum^{j \in \tau_i} | < j, i >|) + E_t \sum^{j \in \tau_i} | < i, j >|}{t} + \frac{\sigma(\sum^{j \in \tau_i} | < i, j > | d_{i,j}^2)]}{t} . \tag{20}$$

Thus, the charging request sending time T_i of n_i can be predicted as:

$$T_i = T_i^{'(f)} + \frac{C_{(n)} - \Theta}{q_i} . \tag{21}$$

Here, $T_i^{'(f)}$ denotes the finishing time of n_i's latest charging service, which can be measured by monitoring WCV's moving activities.

Once the level of any node falls below the energy threshold, the charging request $<i, x_i, y_i, q_i, T_i>, (i \in N)$ with full information inside is able to be predicted in advance.

Above process enables the attacker to tamper a charging request, then we will demonstrate how to mount an efficient DoC attack to destruct network functionality.

B. CoDoC Attack Algorithm

In this section, we present the details of our CoDoC attack algorithm. The attacker is able to manipulate malicious nodes to send fake charging requests with modified requesting time as well as false energy consumption rate in order to cheat WCV. As a result, if the WCV is not aware of the happening of such an attack, and arranges the charging sequence as usual, some nodes will be unable to obtain service from WCV before exhausting their energy. In that case, emergency nodes will potentially run the risk of exhausted unintentionally by this intentional charging attack. Even worse, a PoI is likely to be missed when all sensors who cover that PoI all run out of power (see Equation (8)). With the strong positive correlation between PoIs and sensor nodes, the problem can be transformed into: maximizing the number of dead nodes. Hence, the transformed problem can be divided into three independent subproblems according to the configurable parameters from the view of an adversary.

- A_1: How many fake charging requests should be added into $Re^{(m)}$ set?
- A_2: Which moment $T^{(m)}$ is the best time for sending fake requests?
- A_3: How should the energy consumption rate $q^{(m)}$ be maliciously set in the proposed fake charging request?

Exactly, a sensor node will exhaust its own energy if and only if much time are spent on serving others' charging requests before itself takes part in the decision-making process and becomes the highest-priority node. However, constrained by Reachable-in-Time Test in Equation (2), a sensor node might exhaust energy because WCV is not able to reach the sensor before its charging deadline. Therefore, the condition of n_i's death in Equation (9) can be further computed as:

$$Dead_i = \begin{cases} 1, & \Delta T_i + \sum_{k=0}^{|\Psi_i|} t_k^{(s)} \geqslant \frac{\Theta}{q_i} - \frac{D_{i-1,i}}{v} \\ 0, & \Delta T_i + \sum_{k=0}^{|\Psi_i|} t_k^{(s)} < \frac{\Theta}{q_i} - \frac{D_{i-1,i}}{v} \end{cases} . \tag{22}$$

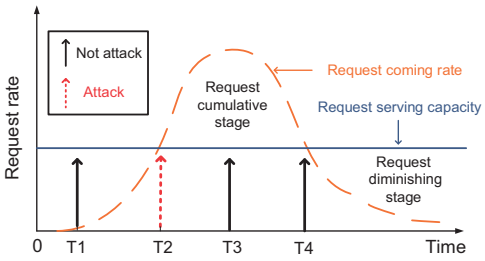


Fig. 3. Request accumulation process of WCV

Here, $|\Psi_i|$ is the length of waiting queue of n_i. $t_k^{(s)}$ refers to the time spending on serving node n_k. $D_{i-1,i}$ is the distance between n_i and the previous serviced node n_{i-1}. ΔT_i denotes the time interval between sending request and firstly taking part in decision-making process. When the request delivered by n_i is received, the WCV has three states: charging, moving, and waiting. Thus, the time interval ΔT_i can be calculated as follows:

$$\Delta T_i = \begin{cases} \frac{C^{(n)} - [\theta - (\frac{D_s}{V} + T_i - T_s) q_i]}{q_r} + \frac{D_s}{V}, & moving \\ T_s + \frac{D_{s-1,s}}{V} + \frac{C^{(n)} - (\theta - \frac{D_{s-1,s}}{V}) q_s}{q_r} - T_i, & charging \\ 0, & waiting \end{cases} . \tag{23}$$

Here, T_s denotes the request sending time of current serviced node n_s. D_s refers to the distance between n_s and WCV. $D_{s-1,s}$ is the distance between n_s and the previous serviced node n_{s-1}. Since the adversary has the knowledge of the network protocol, the current serviced node n_s can be simply confirmed by WCV's location and moving direction.

The total servicing time in Equation (22) can be further calculated as:

$$\sum_{k=0}^{|\Psi_i|} t_k^{(s)} = \sum_{k=0}^{|\Psi_i|} t_k^{(m)} + \sum_{k=0}^{|\Psi_i|} t_k^{(c)} , \tag{24}$$

where $t_k^{(m)}$ and $t_k^{(c)}$ are the time spending on moving to n_k and charging for n_k, respectively. Thus, the constraint of n_i's exhaustion in Equation (22) is divided into total charging time and total moving time.

Besides, to solve subproblem A_1, as sensor node n_i is preempted by m_i, the charging tour and serving time of WCV are prolonged simultaneously, which is depicted in Figure 1. In other words, with the increase of both total moving time and total charging time given in Equation (24), node n_i is more likely to exhaust its energy.

Above all, for any charging queue, each inserted request can make nodes more likely to die. Thus, the adversary tends to send as more fake charging requests as possible, with the residual energy of request sender conforming to the constraints in Equation (13).

To describe the selection process more clearly, we depict the request accumulation process in Figure 3. Before moment $T2$, energy of most nodes are higher than the warning threshold, few requests are gathered by WCV. Therefore, WCV will be standby after finishing charging a node because at that moment the charging queue Ψ is still empty. Later on, the charging

queue becomes longer because requests coming faster than WCV's service capacity. Hence, WCV gathers more requests and starts to service for the accumulated requests after moment $T4$. Thus, Ψ enters a request diminishing stage.

Therefore, the request charging stage at the moment T, denoted as St_T, can be formulated as:

$$St_T = \begin{cases} 1, & |\Psi^{(T)}| < |\Psi^{(T+\Delta T)}| \\ -1, & |\Psi^{(T)}| \geq |\Psi^{(T+\Delta T)}| \end{cases} . \quad (25)$$

Here, $|\Psi^{(T)}|$ denotes the length of queue at the moment T, $|\Psi^{(T+\Delta T)}|$ refers to the length of queue at the next decision making moment, and ΔT denotes the time interval between current time and the next decision making moment. ΔT can be calculated using Equation (23) by simply replacing T_i with T. $St_T = 1$ indicates the request cumulative stage and $St_T = -1$ refers to the request diminishing stage.

1) Scheme analysis I: Based on the aforementioned analysis, only in a request cumulative stage or steady stage, can an inserted request lead to a longer waiting queue $|\Psi_i|$ for an upcoming request (see Equation (22)). Thus, to solve subproblem A_2, the best opportunity for sending fake requests is at the beginning of cumulative stage, i.e., $T^{(m)} = T2$ (see Figure 3). Hence, $T2$ is the earliest moment which satisfies the constraint of $St_T = 1$.

2) Scheme analysis II: To solve subproblem A_3, once a malicious charging request with a higher energy consumption rate $q^{(m)}$ is delivered by m_i, the remaining lifetime will be reduced. Hence, m_i will receive a higher mixed priority and preempt more nodes with low priorities. As a result, the waiting queue $|\Psi_i|$ of more preempted nodes in Equation (24) will be prolonged.

Moreover, a higher energy consumption rate can lead to a longer charging service time $t_i^{(c)}$ (see Equation (4) and Equation (5)). Besides, since the servicing moment T of preempted sensor nodes are further delayed (see Equation (4)), all their charging time will be prolonged. Therefore, the total charging time $\sum_{k=0}^{|\Psi_i|} t_k^{(c)}$ in Equation (24) will be prolonged as well.

Based on above analysis, the higher the malicious energy consumption rate $q^{(m)}$ is set, the longer total servicing time in Equation (24) will be obtained. Therefore, the best malicious energy consumption rate will be set as $q^{(m)} = q_{max}$.

However, in that case, all fake requests will have identical parameters $T_i^{(m)}$ and $q_i^{(m)}$, which will be easy to recognize, to avoid this problem, m_i's delivering moment $T_i^{(m)}$ and fake energy consumption rate $q_i^{(m)}$ are formulated as:

$$T_i^{(m)} = T^{(m)} + \Delta\zeta_t \quad (26)$$

$$q_i^{(m)} = q^{(m)} - \Delta\zeta_q . \quad (27)$$

Here, $\Delta\zeta_t$ and $\Delta\zeta_q$ are distributed randomly in $[0, \zeta_t]$ and $[0, \zeta_q]$ accordingly, where ζ_t and ζ_q denote tiny parameters which make negligible effective to CoDoC attacking algorithm.

3) CoDoC attack algorithm: To maximize the number of exhausted nodes as well as the missed PoIs, we hereby propose CoDoC, a collaborative denial of charge attack for WRSN in Algorithm 2.

Algorithm 2 proceeds as follows. Initially, an adversary calculates how long it will take before making the next decision (Line 1-2). Then, the adversary predicts the request sending time of each sensor node and calculates how many requests will come before the next decision making moment (Line 3-9). If the queue length has been prolonged at the next decision making moment (Lines 10-11), the attacker will launch CoDoC attack by all malicious nodes which satisfy Equation (13) (Line 10-21).

Algorithm 2 CoDoC Attack Algorithm

Input: A malicious node set M, number of charging requests $|\Psi|$
Output: A set of fake charging requests $Re^{(m)}$
1: Calculate the next decision making time ΔT according to Equation (23);
2: $|\Psi^{(T+\Delta T)}| \leftarrow 0$;
3: **for** $i \leftarrow 1$ to $|N|$ **do**
4: Update the predicted energy consumption q_i according to Equation (14) - Equation (20);
5: Update the predicted time of request sending time T_i according to Equation (21);
6: **if** $T_i \in (T, T + \Delta T)$ **then**
7: $|\Psi^{(T+\Delta T)}| \leftarrow |\Psi^{(T+\Delta T)}| + 1$;
8: **end if**
9: **end for**
10: Calculate St_T according to Equation (25);
11: **if** $St_T = 1$ **then**
12: **for** $i \leftarrow 1$ to $|M|$ **do**
13: Test whether m_i satisfies Equation (13);
14: **if** m_i meets Constraints (UBC, RLC, and LBC) **then**
15: $q_i^{(m)} \leftarrow q_{max} - \Delta\zeta_q$;
16: Add $<i, x_i, y_i, q_i^{(m)}, T + \Delta\zeta_t>$ into $Re^{(m)}$;
17: $|\Psi^{(T+\Delta T)}| \leftarrow |\Psi^{(T+\Delta T)}| + 1$;
18: **end if**
19: **end for**
20: **end if**
21: **return** $Re^{(m)}$;

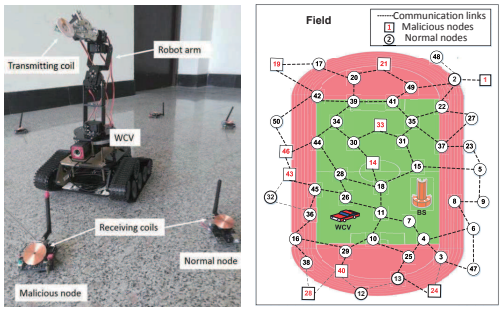
V. Test-Bed Experiments

To demonstrate the effectiveness of our proposed scheme, test-bed experiments based on state-of-the-art on-demand charging architectures (e.g. NJNP [16], mTS [18], and ED-F [17]) are conducted. Detailed configurations are listed in Table I.

TABLE I
Parameters of experiments and simulations

Parameters	Experiments	Simulations
Network size (m^2)	200×100	500×500
Number of nodes	50	250
Coverage radius of node (m)	3	3
Ratio of malicious nodes	0.2	0.2
Number of PoIs	50	50
Node's consumption rate (mJ/s)	$2.5 - 4$	$2.5 - 4$
Traveling consumption rate (J/m)	8	8
Charging consumption rate (J/s)	2	2
Charging efficiency	0.90	0.90
Charging threshold	0.10	0.10
Initial energy of node (KJ)	12	12
Initial energy of WCV (KJ)	500	500
Speed of WCV (m/s)	1	1
Attacking algorithm	DoC, RA, UBA	DoC, RA, UBA

As shown in Figure 4, 50 rechargeable sensor nodes including 10 malicious nodes were randomly deployed in the 200×100 m^2 field with $P = 50$ PoIs. The warning threshold of sensor nodes is 10% and the charging efficiency is 0.9. The size of sensor nodes is $50mm \times 70mm$, with a $12KJ$ battery and a $31mm \times 47.5mm$ receiving coil equipped.



(a) Test-bed: mobile robot arm and two types of rechargeable

(b) Experiment scene: a soccer field

Fig. 4. Test-bed and a soccer field experiment

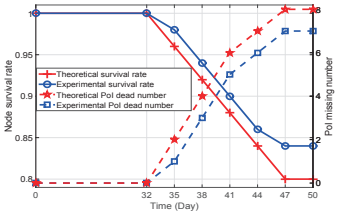


Fig. 5. Comparison between theoretical and experimental results

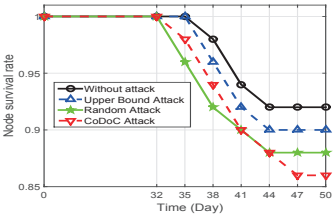


Fig. 6. Comparison of node survival rate when implementing with mTS

First, we compare the differences between theoretical and experimental results on node survival rate and PoI missing number in Figure 5. The results of experiments approximately coincide with theoretical results, which validate the correctness of our theoretical results.

Then, to clarify the effectiveness of CoDoC in a realistic scenario, we compare the performance of CoDoC with two baselines: random attack (RA) and Upper-Bound attack (UBA). RA refers to sending fake requests at a random time, without cooperation between malicious nodes. UBA means to send fake requests as soon as the energy of malicious node reaches the Upper-Bound-Constraint (see Equation (11)), in order to preempt charging queue with the maximum frequency.

As shown in Figure 6, we observe that CoDoC attack leads to 30% and 147% additional dead sensor nodes comparing with UBA and RA. The reason is that CoDoC sends malicious requests more intensively than others in a short time (see Section IV-B1).

Specifically, as shown in Figure 6, we observe that when UBA is implemented, the survival rate drops the most rapidly at the beginning and then remarkably slows down after the 38th day. The reason is that UBA sends fake requests on the maximum frequency without awareness of the upcoming requests. Though a portion of malicious nodes exhaust energy due to the limited capacity of WCV at the beginning, insufficient malicious requests actually preempt the charging queue when normal charging requests arrive.

VI. Simulation Evaluations

To demonstrate CoDoC is also feasible for large-scaled WRSNs, extensive simulations are conducted here. Similarly, we compare our CoDoC attack algorithm with RA and UBA.

At first, we pay attention to node survival rate, indicating the ratio of survival node. We compare CoDoC attack with UBA and RA under three typical on-demand charging architectures: NJNP [16], mTS [18], and EDF [17]. Besides, to illustrate the impact of the CoDoC attack, we also depict the survival rate when no attack is launched. As shown in Figure 7, no matter which charging scheme is taken, we observe that CoDoC attack leads at least 20%, 23%, and 142% additional dead sensor nodes comparing with other algorithms. The reason is that all malicious nodes own the highest priority in the temporal-only charging scheme. Thus, "hungry" nodes need to wait for a long time until malicious nodes are all served.

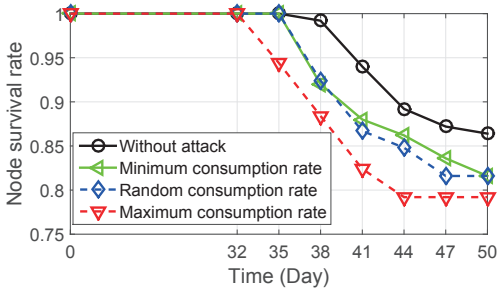


Fig. 9. Comparison of node survival rate under different energy consumption rate when mTS is employed

Moreover, no matter which attack scheme is implemented, the number of missed PoIs increases at the beginning and then converges to a steady state (see Figure 8(a)). The reason is that no matter how malicious nodes send fake requests, the maximum number of coming requests is smaller than WCV's charging capability, therefore, no requests will be abandoned.

Then we demonstrate the advantage of launching a charging attack at the beginning of cumulative stage (see Section IV-B1). As shown in Figure 8(b), the queue length usually firstly increases and then decreases. Specially, the queue length under UBA rises sharply and falls around the 32th day. The reason is that UBA sends fake requests without awareness of the upcoming requests. Thus, there will be a long sending time gap between malicious requests and normal requests. Thus, WCV can make full use of the long time gap (see Equation (23)) to handle malicious requests.

As shown in Figure 8(c), the total service time of CoDoC attack rises simultaneously with the situation of without attack before the 43th day. The reason is that CoDoC attack is not launched until WCV no longer gets rest. Afterwards, CoDoC attack leads to at least 170% additional total service time ultimately. The reason is that more accumulated requests need to be serviced according to Equation (24).

Then we measure the relations between node survival rate and energy consumption rate as depicted in Figure 9. We observe that when consumption rate is maximized, CoDoC attack leads to 47% additional dead nodes. The reason is that a malicious request with maximum energy consumption rate can not only preempt most requests but also prolong their total waiting time (see Equation (24)).

VII. Conclusion

In this paper, we have proposed a novel CoDoC algorithm on WRSN for on-demand charging architecture. At first, we propose a generalized on-demand charging model as the basis.

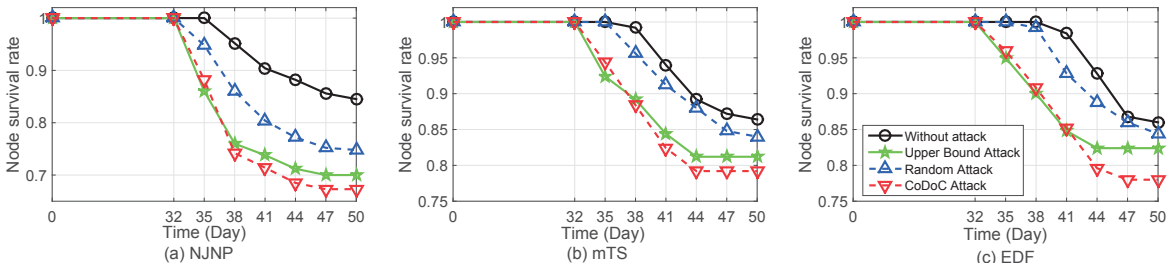


Fig. 7. Comparison of node survival rate when implementing with different charging schemes: (a) NJNP, (b) mTS, and (c) EDF

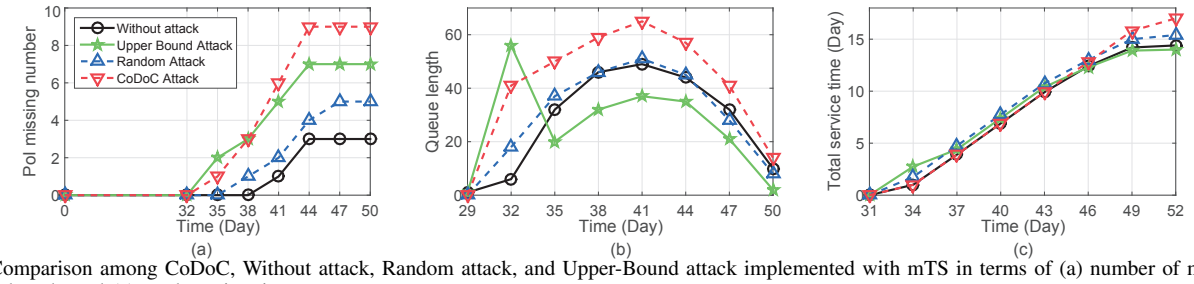


Fig. 8. Comparison among CoDoC, Without attack, Random attack, and Upper-Bound attack implemented with mTS in terms of (a) number of missed PoI, (b) queue length, and (c) total service time

Then a request prediction method (RPM) is introduced for predicting the emergence of charging request. Afterwards, CoDoC attack algorithm is developed, through which network destruction can be maximized. Finally, to demonstrate the outperformed features of CoDoC, extensive simulations and test-bed experiments are conducted. The results show that CoDoC leads to 20% to 142% additional dead sensor nodes in different on-demand architectures, comparing with BA and UBA algorithms.

In the future, we will focus on the performance of CoDoC taken with multiple WCVs. Moreover, the computational overhead and complexity of CoDoC attack will also be explored.

ACKNOWLEDGMENT

This research is sponsored in part by the National Natural Science Foundation of China (61872052, 61602080, 61772113, 61733002, 61842601), National Key Research and Development Program (2017YFC0821003-2), and the "Xinghai Scholar" Program in Dalian University of Technology.

REFERENCES

[1] Wan Du, Zhenjiang Li, Jansen Christian Liando, and Mo Li, "From rateless to distanceless: Enabling sparse sensor network deployment in large areas," in *Proceedings of ACM SenSys*, 2014, pp. 134–147.

[2] Zhidan Liu, Zhenjiang Li, Mo Li, Wei Xing, and Dongming Lu, "Path reconstruction in dynamic wireless sensor networks using compressive sensing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 4, pp. 1948–1960, 2016.

[3] Yuanyuan Yang and Cong Wang, *Wireless Rechargeable Sensor Networks*. Springer International Publishing, 2015.

[4] Tengjiao He, Kwan Wu Chin, and Sieteng Soh, "On using wireless power transfer to increase the max flow of rechargeable wireless sensor networks," in *IEEE ISSNIP*, 2015, pp. 1–6.

[5] Haipeng Dai, Xiaoyu Wang, Alex X. Liu, Huizhen Ma, and Guihai Chen, "Optimizing wireless charger placement for directional charging," in *IEEE INFOCOM*, 2017.

[6] Sheng Zhang, Jie Wu, and Sanglu Lu, "Collaborative mobile charging," *IEEE Transactions on Computers*, vol. 64, no. 3, pp. 654–667, 2015.

[7] Lingkun Fu, Peng Cheng, Yu Gu, Jiming Chen, and Tian He, "Optimal charging in wireless rechargeable sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 1, pp. 278–291, 2016.

[8] Chi Lin, Jingzhe Zhou, Chunyang Guo, Houbing Song, Guowei Wu, and Mohammad S. Obaidat, "TSCA: A temporal-spatial real-time charging scheduling algorithm for on-demand architecture in wireless rechargeable sensor networks," *IEEE Transactions on Mobile Computing*, vol. 17, no. 1, pp. 211–224, 2018.

[9] Wan Du, Zikun Xing, Mo Li, Bingsheng He, Lloyd Hock Chye Chua, and Haiyan Miao, "Sensor placement and measurement of wind for water quality studies in urban reservoirs," *ACM Transactions on Sensor Networks*, vol. 11, no. 3, pp. 41:1–41:27, 2015.

[10] Wan Du, Jansen Christian Liando, Huanle Zhang, and Mo Li, "When pipelines meet fountain: Fast data dissemination in wireless sensor networks," in *Proceedings of ACM SenSys*, 2015, pp. 365–378.

[11] Haipeng Dai, Yunhuai Liu, Guihai Chen, Xiaobing Wu, Tian He, Alex X. Liu, and Yang Zhao, "Scape: Safe charging with adjustable power," *IEEE/ACM Transactions on Networking*, vol. 26, no. 1, pp. 520–533, 2018.

[12] Quan Chen, Hong Gao, Siyao Cheng, Jianzhong Li, and Zhipeng Cai, "Distributed non-structure based data aggregation for duty-cycle wireless sensor networks," in *IEEE INFOCOM*, 2017.

[13] Peng Ning, An Liu, and Wenliang Du, "Mitigating dos attacks against broadcast authentication in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 4, no. 1, pp. 1:1–1:35, 2008.

[14] Anat Bremler-Barr, Eli Brosh, and Mor Sides, "Ddos attack on cloud auto-scaling mechanisms," in *IEEE INFOCOM*, 2017.

[15] P Tague, D Slater, J Rogers, and R Poovendran, "Vulnerability of network traffic under node capture attacks using circuit theoretic analysis," in *IEEE INFOCOM*, 2008, pp. 161–165.

[16] Liang He, Linghe Kong, Yu Gu, Jianping Pan, and Ting Zhu, "Evaluating the on-demand mobile charging in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 14, no. 9, pp. 1861–1875, 2015.

[17] John A. Stankovic, Marco Spuri, Krithi Ramamritham, and Giorgio C. Buttazzo, *Deadline Scheduling for Real-Time Systems*. Springer Publishing Company, Incorporated, 2013.

[18] Chi Lin, Zhiyuan Wang, Jing Deng, Jiankang Ren, and Guowei Wu, "mTS: Temporal and spatial collaborative charging for wireless rechargeable sensor networks with multiple vehicles," in *IEEE INFOCOM*, 2018, pp. 99–107.

[19] Yuanchao Shu, Hamed Yousefi, Peng Cheng, Jiming Chen, Yu Jason Gu, Tian He, and G. Shin Kang, "Near-optimal velocity control for mobile charging in wireless rechargeable sensor networks," *IEEE Transactions on Mobile Computing*, vol. 15, no. 7, pp. 1699–1713, 2016.

[20] B. Parno, A. Perrig, and V. Gligor, "Distributed detection of node replication attacks in sensor networks," in *IEEE SP*, 2005, pp. 49–63.

[21] Tamara Bonaci, Linda Bushnell, and Radha Poovendran, "Node capture attacks in wireless sensor networks: A system theoretic approach," in *IEEE CDC*, 2010, pp. 6765–6772.

[22] Songyuan Li, Lingkun Fu, Shibo He, and Youxian Sun, "Near-optimal co-deployment of chargers and sink stations in rechargeable sensor networks," *ACM Transactions on Embedded Computing Systems*, vol. 17, no. 1, pp. 10:1–10:19, 2017.