Towards a Practical Energy Conservation Mechanism with Assistance of Resourceful Mules

Yi Qu, Ke Xu, *Senior Member, IEEE,* Jiangchuan Liu, *Senior Member, IEEE,* and Wenlong Chen

***Abstract*—As Wireless Sensor Networks (WSNs) gradually move from specialized fields such as military and industry towards domains with general purposes, more and more sensors locate around our living areas. The reality that various wireless devices coexist in new circumstances encourages us to come up with new ideas to solve the extremely energy-constrained problem in WSNs. In this paper, we propose Energy Conservation with Assistance of Resourceful Mules (ECARM), a mechanism that opportunistically utilizes Resourceful Mules (RMs) such as specifically-designed powerful sensors or ubiquitously-used laptops, tablet PCs and smart phones to act as assistants and save energy for WSNs. We verify ECARM through extensive simulations written on the OMNET++ platform. Single RM simulations show that 43% sensors in an RM's communication range enjoy power reduction by decreasing their wake-up time to 16% at most. Multiple RM simulations illustrate that 86% sensors in the simulated network benefit from 14 RMs, and wake-up time of 56% sensors decrease to 50% below. We emphasize that ECARM can also be applied in duty-cycled WSNs that adopt schemes such as ContikiMAC and X-MAC. Simulation results demonstrate that the duty-cycling ratio of ContikiMAC is further decreased by at least 20.9% after the ECARM application.**

***Index Terms*—Wireless Sensor Network (WSN), resourceful mule, energy conservation, duty-cycling.**

I. Introduction

In the last decades, Wireless Sensor Networks (WSNs) have attracted lots of interests in research and industrial communities. Numerous applications including military assistance [1], intelligent transportation [2], environment surveillance [3] etc. have been developed and they will inevitably affect most aspects of our lives. However, the severe energy constraint is the key factor that hinders the wide-scale deployment of WSNs [4] [5] [6]. This constraint is mainly caused by three factors: (1) Sensors are mainly battery-powered [7]; (2) Sensors are always left unattended after deployment [8]; (3) A WSN always consists of a great number of sensors, such that battery replacement could consume intensive labor.

Fortunately, the new development trend of WSNs provides us opportunities to overcome these limitations. In recent years, lots of WSN applications have shifted from specific fields to domains related to people's daily lives such as health care [9] [10] and intelligent home [11], where a variety of wireless devices coexist. Compared to tiny sensors, mobile devices such as smart phones and laptops have abundant energy and powerful computation capacities. For simplicity, we use the term Resourceful Mules (RMs) to represent these mobile devices. Leveraging RMs to act as assistants to save energy for weaker sensors has become a new research hot spot [12] [13] [14] [15] [16] [17] [18].

Generally, there are two kinds of methods that utilize motionless and movable RMs, respectively. Motionless methods (e.g., cluster-head [18]) assume that RMs are part of WSNs and their locations are fixed within entire lifetimes, which can be applied only if RMs are deployed before the start of WSNs. In this paper, we focus on movable RM cases.

Actually, movable RMs can both have mobile and static characteristics. Mobility accounts for the most important place when RMs shift from one place to another, while static characteristics become dominant if RMs sojourn. However, most researches focus on the mobile aspect of movable RMs and little attention is paid to explore their benefits in static situations. We give an example to illustrate the difference. Suppose a WSN is deployed in an office building and p_1 and p_2 are two smart phones carried by two employees P_1 and P_2 who work there. Every sensor keeps awake and periodically monitors the environments. Whenever sensory data are gathered by sensors, they will be forwarded via multiple wireless hops to a sink. The movement pattern of the two employees are distinct: P_1 is a supervisor and keeps traversing the office area while P_2 is a worker who enters the office, sits down statically and keeps working for hours.

The existing methods [12] [13] [14] [15] [16] [17] [18] utilize mobile RMs such as p_1 as follows: When p_1 passes by a sensor, it collects data from the sensor. After p_1 moves close to the sink, all the collected data are transmitted to the sink. Sensor energy is saved because sensory data are forwarded by physical movements of p_1 rather than costly multi-hop wireless transmissions. This type of methods requires predefined movement pattern of RMs and inevitably causes huge delay (tens of minutes) since physical movement is much slower than wireless transmission.

Since most employees (mobile devices) in an office building sojourn statically for more than 8 hours every day, why don't we incorporate them in the WSN to improve the performance of weaker sensors? In this paper, we exploit benefits from long-been-neglected static aspects of movable RMs such as p_2. However, extracting energy gains through static RMs entails a number of research challenges: (1) RMs with long sojourn

Yi Qu and Ke Xu are with the Department of Computer Science & Technology, Tsinghua University, Beijing 100084, China (e-mail: quy11@mails.tsinghua.edu.cn; xuke@tsinghua.edu.cn).

Jiangchuan Liu is with the School of Computing Science, Simon Fraser University, 8888 University Drive, Burnaby, British Columbia, Canada V5A 1S6 (e-mail: jcliu@cs.sfu.ca).

Wenlong Chen is with the College of Information Engineering of Capital Normal University, Beijing 100084, China (e-mail: cwl@csnet1.cs.tsinghua.edu.cn).

TABLE I
RADIO POWER CHARACTERIZATION

Radio mode	Power consumption (mW)	
	Atheros	TR1000
transmission(tx)	127	14.88
reception(rx)	223.2	12.50
idle	219	12.36
sleep	10.8	0.016

time need to be selected out from lots of random RMs. (2) The proposed mechanism requires distributed nature since each RM has little knowledge of other RMs. (3) Sensor behaviors need to be coordinated appropriately such that a part of them could switch to the low power sleep mode and save energy.

To address these challenges, this paper presents a practical mechanism named Energy Conservation with Assistance of Resourceful Mules (ECARM). We analyze typical dynamics of employees in an office building and find that if an employee has already sojourned in a spot for several minutes, he/she has a high probability to continue for a long duration. Then, we design a threshold-based method to select out those RMs with potential benefits. After an RM is incorporated, sensors within its coverage are classified into two types: source and forward nodes. Source nodes reelect the RM as their new next-hop and send data to it. The RM alternatively chooses a forward node to relay the collected data. As long as there are more than one forward node, the remaining forward nodes could stay in the sleep mode and conserve energy. Although only forward nodes benefit from RMs, sensor types are not fixed. In a long term, source nodes will become forward nodes and enjoy energy saving, too.

The above situation assumes that every sensor keeps awake continuously before RM participation, which is simple but energy consuming. Nowadays, duty-cycling schemes such as ContikiMAC [19] and X-MAC [20] are prevalently adopted. One of the most important advantages of ECARM is that it can be applied even sensors are already duty-cycled. After an RM is incorporated, source nodes send data to the RM directly without long preamble or repeated transmission, while forward nodes experience dual cyclings. As a result, both types of sensors enjoy energy saving and energy conservation efficiency of ECARM is increased.

The main contributions of our work can be summarized as follows:

- We propose a practical mechanism ECARM, which exploits benefits from long-been-neglected static aspects of movable RMs.
- Through extensive simulations, we show that 43% sensors in an RM coverage enjoy power reduction and their lifetime is prolonged by a factor of 6.25 in the best case.
- We also apply ECARM in a duty-cycled WSN with state-of-the-art ContikiMAC [19]. Simulation results demonstrate that the duty-cycling ratio of sensors within RM's coverage is further decreased by 20.9% at least, which is a significant improvement over pure ContikiMAC.

The remainder of the paper is organized as follows: Section II presents the ECARM overview and the design details are described in Section III. Section IV shows the simulation and we apply ECARM in a duty-cycled WSN in Section V. Some practical issues are discussed in Section VI. Related work is described in Section VII. Finally, Section VIII concludes the paper.

II. OVERVIEW

First of all, we explain why wake-up/sleep cycling is an effective way for energy saving (here we use the term wake-up/sleep cycling to differentiate from duty-cycling schemes in the MAC layer such as ContikiMAC [19] and X-MAC [20]). As illustrated in [4] [21], radio is the main energy consumer of a sensor. More specifically, radio power consumption in transmission (tx), reception (rx) and idle modes is roughly at the same level and orders higher than the sleep mode, whose energy consumption is negligible [21]. The energy profile of the Atheros card [22] and the TR1000 radio [4] (see Table I) are examples of this regularity. Therefore, wake-up/sleep schedules that put some sensors into the sleep mode are efficient energy-saving methods.

Secondly, we note that ECARM is suitable for scenarios with limited RM dynamics. Offices and libraries are typical applicable environments, where people (RMs) spend most of the time staying in a spot statically and seldom move. Whenever they move, they quickly leave that spot. Another requirement of ECARM is that routing protocols adopted by WSNs should ensure a structural topology. For example, shortest-path or tree-based (e.g., CTP [23]) routing protocols are appropriate for ECARM. (These two types are most prevailing routing protocols in WSNs.) However, random routing schemes such as [24] are inappropriate.

Then, we describe the main steps of ECARM: RM selection, information exchange, cycling coordination and fast recovery.

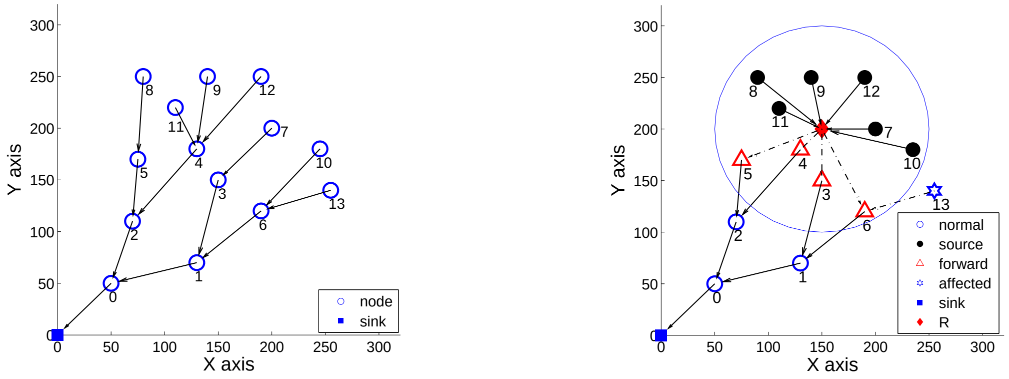
RM Selection. The main purpose of this step is to find those RMs with a high probability to sojourn statically for a long duration. After an RM R emerges in WSNs, its neighbors will learn its existence through lower-layer neighbor discovery protocols. The sojourn time of R will be recorded and only after it passes a predefined threshold K_0 (to be discussed in Subsection III-A), R will be allowed to participate in WSNs and become an assistant.

Information exchange. Before R coordinates sensor behaviors, necessary information is required, which includes all R's neighbors and their corresponding next-hop nodes. In Fig. 1, R's neighbor set $N_R = \{3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$, its neighbors' next-hop set $N_R^x = \{1, 2, 2, 1, 3, 5, 4, 6, 4, 4\}$. N_R can be obtained directly through neighbor discovery, and N_R^x can be piggy-backed in the neighbor discovery packets.

Cycling coordination. With adequate information, R classifies all its neighbors into two categories: source nodes and forward nodes. In our example, source node set $N_R^s = \{7, 8, 9, 10, 11, 12\}$ and forward node set $N_R^f = \{3, 4, 5, 6\}$. Then, R figures out wake-up/sleep schedules of forward nodes and results are broadcasted to all R's neighbors. (Details are described in Subsections III-B and III-D.)

After all R's neighbors are assigned with new roles, they adjust behaviors as follows: source nodes reelect R as their

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2014.2371056, IEEE Internet of Things Journal



(a) Original topology

(b) WSN topology after RM participation

Fig. 1. An example of ECARM.

next-hop and send data to R, and forward nodes begin wake-up/sleep cycling under the coordination of R. Energy of forward nodes are saved because only one of them is required to be awake, the remaining ones could stay in the sleep mode. As to source nodes, energy consumption remains unchanged. However, after we apply ECARM in a duty-cycled WSN adopting X-MAC[20] or ContikiMAC[19], source nodes could send data to RMs without long preamble. Thus, energy is also saved.

Fast Recovery. RMs will not sojourn forever and may leave the WSN field. Whenever sensors find that R has disappeared, they are obliged to return to its original behavior as quickly as possible. We define a threshold K_1. If K_1 data packets of source nodes are lost consecutively or K_1 wake-up durations of forward nodes do not receive any packet from RMs, sensors recognize that R has left. Then, they delete R from their neighbor table and return to the connection relationships before R appears (see Fig. 1(a)). Through extensive simulations, we find that $K_1 = 3$ obtains nice tradeoff between recovery speed and detection reliability.

III. Design

In this section, we illustrate ECARM details, which is designed under the principle of effectiveness, low-complexity and robustness.

A. Threshold K_0

The optimal setting of K_0 is highly related to the mobile regularity of RMs (persons). Here, we give an example to show how to find its proper value. We analyze a dataset collected by Massachusetts Institute of Technology (MIT) Reality Mining Group, which contains the dynamics of 23 employees at a Chicago-based data server configuration firm for one month [25]. We first extract the locations where an employee has stopped by, and then we record the corresponding sojourn time at each location. The results are shown in Fig. 2. We find that the majority of sojourn time is short (no greater than 1 min), however, if an employee has already stayed at a location for minutes, he/she tends to stay for a longer duration. Suppose the MIT dataset are collected in an area covered by a WSN and every stop/movement behavior of an employee stands for the sojourn/leave mobility of an RM, a threshold $K_0 = 4\ mins$ will filter out 96% RMs and the remaining 4% with sojourn times range from 4 mins to 271 mins (the average value is 13 mins) will be recognized as useful assistants to the WSN.

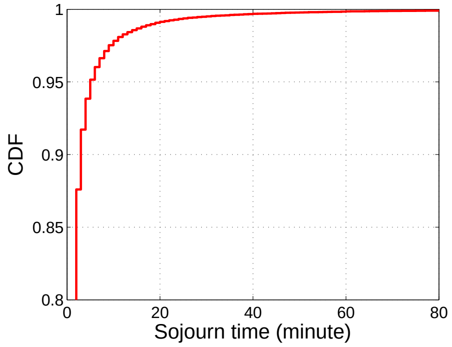


Fig. 2. Set proper K_0 based on RM mobile regularity.

Based on the observation that sensors are tiny devices with very limited capacity (e.g., computation, storage and power). Let sensors themselves to decide how to function optimally based on the movement patterns of RMs will consume lots of sensor capacities, which is undesirable. In fact, involvement of system parameters can be effective to offload some stressful tasks from tiny sensors to powerful PCs. In this paper, we record and analyze movement patterns of potential RMs and get the optimal settings of parameter K_0. The MIT dataset can be deemed as a typical application scenario, and K_0 = 4 mins will be suitable for most cases. However, if this value is improper for some special scenarios, only one person, one watch and several hours of observation will enough to find a proper K_0. After that, optimal values are adopted by all sensors. The proposed solution is simplified and only focuses on its main task, i.e., energy conservation.

B. Node Classification

We borrow ideas from [6] [26], which classify nodes by their connections. In GeRaF [6], sensors in the coverage area

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2014.2371056, IEEE Internet of Things Journal

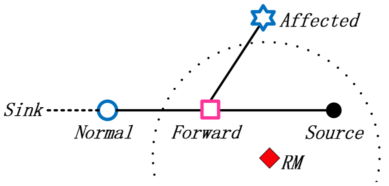


Fig. 3. Sensors are classified by their connection relationships.

of source nodes are partitioned into several parts based on their distance to the sink. Source nodes prefer to choose sensors located in parts closer to the sink as data relays. However, this location awareness assumption is impractical in most WSN application scenarios.

In this paper, such assumption is avoided. As to information collection WSNs, which is the most prevailing WSN application scenarios, all of the sensory data are relayed from data generators toward a sink, such that the sensor's next-hop is closer to the sink than the sensor itself theoretically. Accordingly, data passing through an RM's coverage area are relayed by sensors whose next-hop is both out of that area and closer to the sink than the RM. Under this concept, we classify sensors into four types (for simplicity, in this section, we only consider the situation that each sensor is covered by at most one RM. More complicated situations are discussed in Subsection III-E):

- *Source node*: A node and its next-hop are both neighbors of the same RM.
- *Forward node*: A node is a neighbor of an RM, and its next-hop is not a neighbor of that RM.
- *Affected node*: A node's next-hop is a forward node of an RM. But the node itself is not a neighbor of that RM.
- *Normal node*: A node satisfies that neither itself nor its next-hop is the neighbor of RMs. Without RM, every sensor is regarded as normal.

An example of node classification is given in Fig. 3.

After R's participation, source and forward nodes change their behaviors as described in Section II. Here we emphasize on affected nodes. Forward nodes may be next-hops of sensors that locate out of the communication range of the RM. Without precise cycling information, an affected node A may send data to its next-hop F (belonged to forward nodes) during F is asleep, which will cause serious packet loss. To avoid this, A is obliged to learn the wake-up/sleep cycling of F. (The interaction will be described in detail in Subsection III-C). Then, A buffers packets during F is asleep and sends the buffered packets after F wakes up.

C. Wake-up/sleep Cycling Determination

Within the coverage of R, only one awake forward node is enough for data relay. A time-slot based circular schedule method (with time slot t_s) is adopted because of its simplicity and efficiency. Parameter t_s determines the basic wake-up duration of a forward node. Let n_R denote the number of forward nodes associating with R.

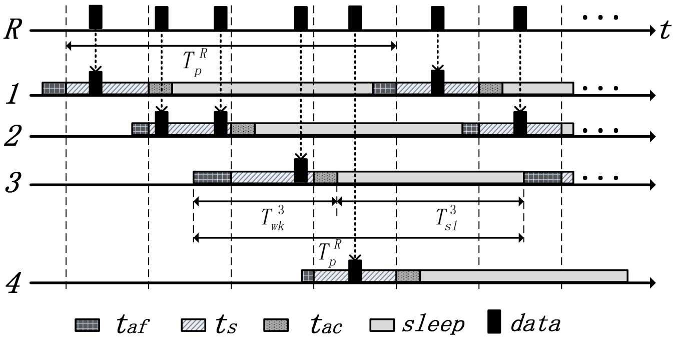


Fig. 5. Time-slot based circular schedule. R relays data to forward nodes 1-4 alternately. At the beginning, Node 1 wakes up and relays data for R, and then it goes to sleep for energy saving and the data relay task is passed to Node 2 and so on.

As to forward node i, where $i = 1, 2 ... n_R$, the time-slot based circular schedule period T_p^R is also its wake-up/sleep cycling period, which is

$$T_p^R = n_R \times t_s = T_{sl}^i + T_{wk}^i \tag{1}$$

where T_{sl}^i and T_{wk}^i are wake-up and sleep durations of i in a single wake-up/sleep cycling.

Different from t_s that is allocated for relaying R's collected data, time interval t_{ac} is designated for the channel access delay caused by the MAC layer. Without t_{ac}, data received at the end of the T_{wk}^i period may not have the opportunity to be transmitted in this cycling, which may cause severe delay especially in the case where t_s is very long. Both t_s and t_{ac} are predefined parameters and identical for all sensors.

We have already mentioned that forward node i may be the next-hop of affected nodes, where data buffered will be transmitted when i wakes up. This part of data may interfere with data from R. To avoid collision, we allocate t_{af}^i to deal with buffered data uniquely. Therefore, we have

$$T_{wk}^i = t_{af}^i + t_s + t_{ac} \tag{2}$$

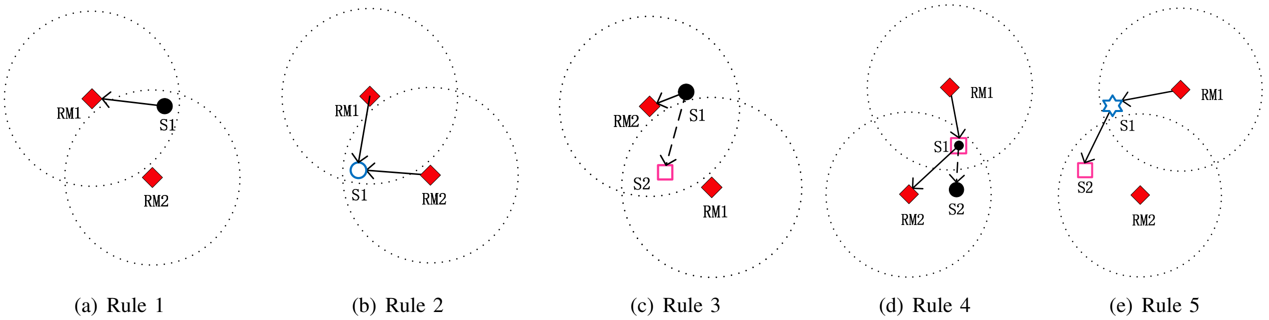
An example of the time-slot based circular schedule where $N_R = 4$ is illustrated in Fig. 5.

Next, we show how to calculate t_{af}^i properly while keeping the simplicity of ECARM. As we already knew, t_{af}^i is designated to deal with data buffered in affected nodes, such that t_{af}^i is determined by the potential number (denoted as P_a^i) of affected nodes and their data rates. We notice that P_a^i is related to the neighbor divergence between i and R. If all neighbors of i locate in R's coverage area, no affected node would exist and t_{af}^i is totally unnecessary. On the contrary, if most neighbors of i locate far away from R, P_a^i becomes larger and a longer t_{af}^i is required. Therefore, we have

$$P_a^i = 1 - \frac{n_{re}^{iR}}{n_{ne}^i} \tag{3}$$

where n_{ne}^i means the number of i's neighbors, and n_{re}^{iR} stands for the number of nodes that are neighbors of both i and R.

We assume that the data collection rate of R is γ_R, and the number of source nodes in R's range is n_R^s. Therefore, the average data rate of source nodes is $\frac{\gamma_R}{n_R^s}$. J. Luo et al. [26] proved that the data rate of sensors in information collection WSNs is a function of its distance from the sink. Hence, data



(a) Rule 1 (b) Rule 2 (c) Rule 3 (d) Rule 4 (e) Rule 5

Fig. 4. The first role is assigned by RM1, after the emergence of RM2, sensors change their behaviors as follows: (a) source node S1 remains unchanged; (b) S1 becomes a normal node after RM1 and RM2 both assign it with forward roles; (c) affected node S1 (its next-hop is forward node S2) becomes source node; (d) forward and source roles coexist on S1; (e) forward node S1 (its next-hop is normal node S2) becomes affected node after S2 is assigned with a forward role.

rates of affected nodes and source nodes corresponding to the same RM are similar. We assume i's data reception rate during t^i_{af} equals to that during t_s, which can be represented by:

$$\gamma_R = \frac{\frac{\gamma_R}{n^s_R} \times T^R_p \times P^i_a}{t^i_{af}} \tag{4}$$

Thus t^i_{af} can be estimated.

D. Interactions and localized synchronization

Node classification and wake-up/sleep cycling results are broadcasted to R's neighbors. After that, source nodes change their next-hops to R instantly. Nevertheless, forward nodes delay wake-up/sleep cycling operations by θ (they keep awake during θ), which is designated for discovering affected nodes. Suppose that each sensor generates data in a Poisson process with parameter λ (other distributions can be handled similarly), which means the interval τ between any two packets is distributed exponentially. If there exists an affected node A, we can assure that its next-hop F will receive data from A (i.e., successfully find A) with a minimum probability of $1-\eta$ after we delay F by θ, where η satisfies that the probability $P\{\tau \geq \theta\} \leq \eta$. In this paper, we set $\eta = 1\%$ (i.e., $\theta = 4.6\lambda$), which means that the lower bound probability to successfully find A is 99%. This is because A's data rate is far bigger than λ considering that it not only generates data itself but also relays data for others. F notifies A of its wake-up/sleep cycling after A is successfully detected. On the contrary, if F only receives data from its corresponding RM during θ, it deems that no affected node exists.

It is easy to find that time-slot based schedules require synchronization among R and its associated forward nodes (if there exist affected nodes, synchronization is extended to two hops). Notoriously, synchronization is an issue that we should avoid due to its complexity and high overhead. However, in ECARM, this issue is simplified to a great extent because of two factors. The first one is that we only require localized synchronization, i.e., at most two hops. As shown in Fig. 5, the boundary of time-slot t_s should be aligned with R's clock. Therefore, whenever R has data to relay, the designated forward node is awake. This is assured by t_{af} and t_{ac} before and after t_s, such that the robustness of the time-slot based circular schedule is increased and small time skew will not influence normal communications, which is the second factor. Synchronization in this paper neither requires high accuracy nor wide scale. Therefore, simple synchronization schemes such as FTSP [27] and TPSN [28] can be adopted.

E. Multiple RMs

The above subsections only consider the situation that each sensor is covered by at most one RM. Here we discuss the multiple-RM situation exclusively. The RM participation indeed brings power conservation for sensors. But it does not mean the more RMs the better. Excessive RMs would cause unnecessary interaction burden and worsen the channel environment. Our policy is that only one RM is allowed to participate in the WSN within its coverage. We adopt a conservative strategy that an RM will ask for participation only after it confirms its uniqueness in its range. If an RM has already joined the network, the newly appeared one will keep silent until the former one disappears.

However, a sensor overlapped by two or more RMs that cannot hear each other will be assigned with multiple roles (types), which may cause sensor behavior confusion. Although this situation rarely happens, the results would cripple WSNs seriously. To ensure sensors function orderly and keep ECARM simple and robust, we set the following rules:

1) Source nodes do not change their behaviors if another source role is assigned.
2) Forward nodes become normal if an additional forward role is assigned.
3) Nodes with affected and source roles at the same time become source nodes.
4) Source and forward roles (only one forward role) coexist.
5) Nodes with affected and forward roles at the same time become affected nodes.

Rule 1 is evident and Rule 2 avoids the situation that multiple asynchronous wake-up/sleep cyclings coexist on a single forward node. We set Rule 3 because source role means sensors could transmit data to always-awake RMs, which is preferable than buffering them and delay-sending to forward nodes. As to Rule 4, sensors with forward and source roles at

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2014.2371056, IEEE Internet of Things Journal

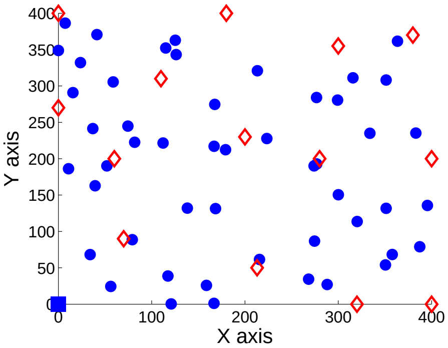


Fig. 6. Location of sensors and RMs. Solid circles, hollow diamonds and the square are sensors, RMs, and the sink, respectively.

the same time both perform wake-up/sleep cycling and send data to RMs. As we discussed above, affected nodes adjust their behaviors based on the cycling of their next-hops. They might miss the chance to send buffered data if they go to sleep. Rule 5 is set to prevent this. Examples are shown in Fig. 4. Based on the above five rules, sensors adjust their behaviors whenever a new role is assigned or an old role is outdated. Algorithms can be easily implemented under the above rules. Here we omit them due to limited space.

IV. Evaluation

We evaluate ECARM through simulations written on the OMNET++ platform [29], a C++ based discrete event simulation tool. We build a simplified protocol stack with 4k line of C++ code [1]. Our simulation setup includes a wireless link with 250 kbps transmission/reception rate and a CSMA-type MAC which complies with the IEEE 802.15.4 standard [30]. Above the MAC layer, we choose a shortest path routing protocol. In our simulation, sensors generate sensory data packets with a length of 100 bytes in a Poisson distribution manner with parameter λ. We have also tried other distribution manners and inhomogeneous generating rates. However, all the simulation results are similar. Data generating methods have limited impact on the final results. Thus we adopt a homogeneous assumption for simplicity. All of the generated data will be transmitted to a unique sink through multi-hop wireless links. The transmission range (denoted as r) of sensors (the sink) are identical to RMs.

As we mentioned in Section II, the energy cost of wireless radios in transmission, reception and idle mode is roughly at the same level and almost an order higher than the sleep mode [21]. Without ECARM, sensor radios are running in always-on state. After the participation of RMs, a part of sensors could sleep opportunistically and their energy consumption is reduced. In this paper, we quantify ECARM's energy conservation efficiency as the sleep ratio (or wake-up ratio), which is the ratio between the cumulative sleep (wake-up) time and the total simulation time.

To comprehensively assess ECARM, we set up two groups of simulations: single RM and multiple RMs. We also adopt two node distributions: random (N sensors distributed in a uniformly random fashion over a rectangle field of size $L \times L$, as shown in Fig. 6) and controlled distribution. (The locations of sensors/sink and RMs are predefined, as shown in Fig. 1(b). Some default parameter settings are shown in Table II. For simplicity, we represent the density of the network as sensor's average number of neighbors, denoted by parameter ρ, where $\rho = N\pi\frac{r^2}{L^2}$.

TABLE II
Simulation parameter settings

Parameters	K_0	K_1	N	r	L	λ
Settings	4 mins	3	50	100	400	10 s

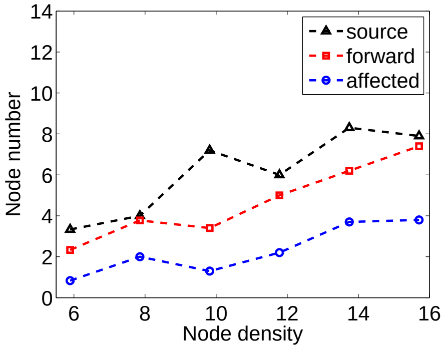


Fig. 7. Number of 3 types of nodes vs. node density.

This section is organized as follows: firstly, only one static RM is involved in the simulated WSN and we focus on the wake-up/sleep cycling part of the proposed solution. Then, we include multiple RMs to evaluate our solution in an ideal application scenario, i.e., all RMs are motionless. Lastly, a group of more practical trace-based simulations are demonstrated with 14 mobile RMs.

A. Single RM

1) Number of 3 types of nodes: Firstly, we care about the number of nodes with source, forward and affected types after the participation of an RM. We locate an RM in the center of the randomly-distributed WSN to ensure all neighbors of the RM are contained in the network. At each density degree, we repeatedly run the simulation at least 10 times and calculate the average number of 3 types of nodes. For each run, we set a different random number seed to obtain different distributions. The results are shown in Fig. 7. On average, within RM's coverage, 57% sensors are source nodes and the remaining 43% are forward nodes. Out of RM's coverage, the average number of affected nodes is of 20.4% node density.

2) Energy conservation vs. number of forward nodes: As we mentioned before, a time-slot based schedule was adopted by ECARM. Therefore, the more forward nodes, the longer sleep duration and the higher sleep ratio. To verify this, we add/remove forward nodes in the controlled distribution topology in the second simulation (with a simulation time of 10000 s) and show the wake-up period T_{wk}, the sleep period T_{sl} and the sleep ratio of Node 6 in Fig. 8. (The results are

[1] The source code of our simulations can be found at http://sourceforge.net/projects/ecarm/?source=directory

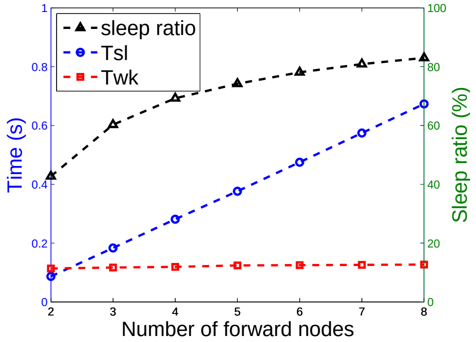


Fig. 8. Forward node: wake-up/sleep cycling vs. node number ($t_s = 100ms$).

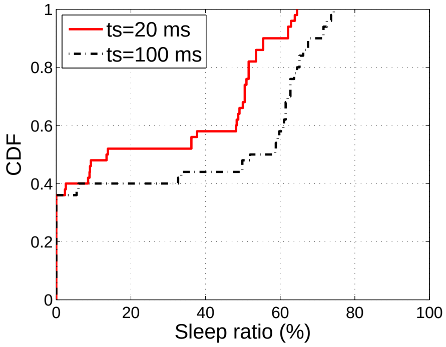


Fig. 12. CDF of sensor sleep ratio in a static scenario.

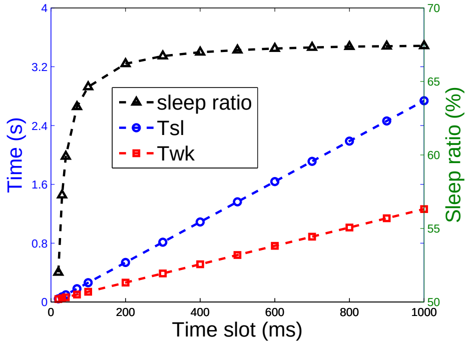


Fig. 9. Sleep ratio of forward node vs. t_s ($n = 4$).

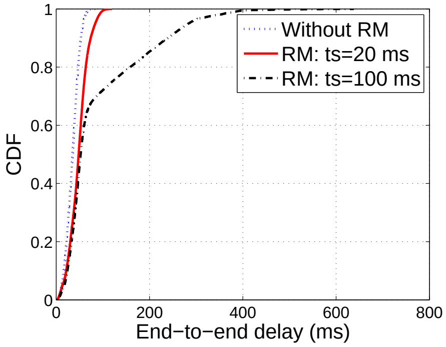


Fig. 13. CDF of packet end-to-end delay in a static scenario.

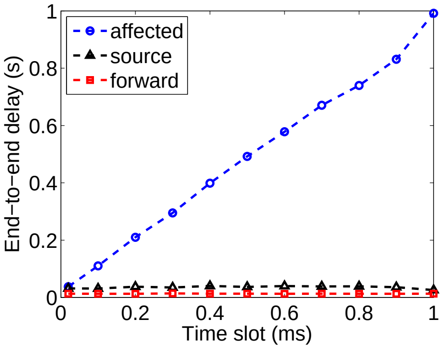


Fig. 10. Average end-to-end delay of data packets from affected, source and forward nodes (R appears at $t = 0s$).

similar to other forward nodes). In the case of 8 forward nodes, Node 6 keeps in the sleep mode for 84% of the total simulation time, which means its lifetime is prolonged by a factor of 6.25 theoretically.

3) Energy conservation vs. time slot t_s: One of the most important parameter of ECARM is time slot t_s. We vary t_s from 20 ms to 1000 ms in the controlled topology and run our third simulation. From Fig. 9, we can see that the sleep ratio of forward nodes increase with t_s. However, it increases slowly after $t_s > 100\ ms$.

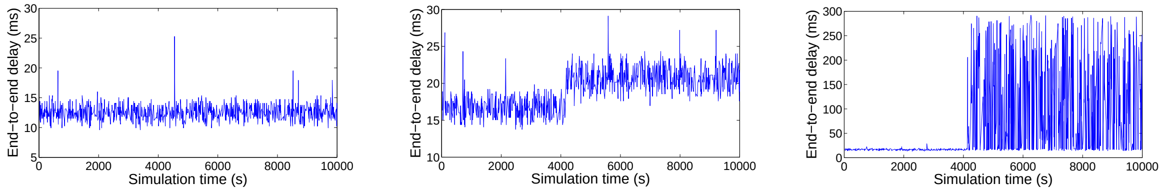
4) Increased end-to-end delay: The above simulations evaluate ECARM's energy efficiency in single RM scenarios. But they did not show its influence (primarily related to increased packet end-to-end delay) to WSNs. To illustrate the delay variation before/after the participation of R, we adjust the appearance time of R from default time $t = 0$ to $t = 4000\ s$ in our fourth simulation. We choose Nodes 6, 12, 13 in the controlled topology to represent forward, source and affected nodes, respectively. (Choosing other nodes will obtain similar results).

Figure 11(a) shows that ECARM does not influence the end-to-end delay of packets generated by forward nodes. This is because whenever sensory information is obtained, forward nodes will wake up (if they are in the sleep mode) and transmit it instantly. As to source nodes, the delay raise is caused by an additional wireless hop, which is approximately 4 ms. For example, in Fig. 1(b), after the participation of R, the relay process of packets from Node 12 is Node 12 $\rightarrow R \rightarrow$ Node 6, rather than Node 12 $\rightarrow$ Node 6 directly. Figure 11(c) shows the packet delay of affected nodes, which increases significantly after the appearance of R. This is mainly caused by those buffered data during the sleep period of its next-hop.

In our fifth simulation, we reset the RM's appearance time to $t = 0$ and vary t_s to show that the average delay of data generated by affected nodes is proportional to the sleep time of their next-hops, which is mainly determined by t_s and the number of forward nodes. (The results are shown in Fig. 10).

The increased end-to-end delays are not only determined by the number of RMs, but also the distribution and connection relationships of sensors/RMs, which is complicated and thus omitted in this paper. However, through extensive simulations, we found that evenly distributing RMs in WSNs to achieve full sensor coverage (i.e., every sensor is covered by at least one

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2014.2371056, IEEE Internet of Things Journal



(a) Forward node (b) Source node (c) Affected node

Fig. 11. The average end-to-end delay of packets generated by affected, source and forward nodes. (An RM appears at time 4000 s and time slot $t_s = 100$ ms.)

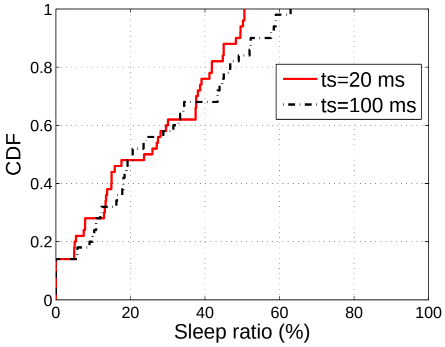


Fig. 14. CDF of sensor sleep ratio in a dynamic scenario.

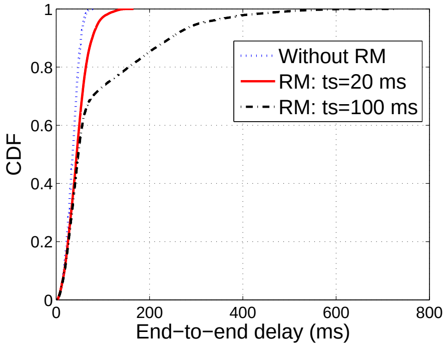


Fig. 15. CDF of packet end-to-end delay in a dynamic scenario.

RM) and reduce redundancy at the same time (i.e., reduce the number of sensors that are covered by more than one RM) is a simple and effective way to get the best tradeoffs between energy conservation and increased end-to-end delay.

B. Multiple RMs

1) Static RM scenarios: Single RM simulations have proved the effectiveness of ECARM. In this part, we evaluate ECARM's energy efficiency in a more practical multiple RM environment. We assume that our laboratory is covered by a WSN and all student laptops are RMs. Sensors are randomly distributed and RMs are evenly located in the WSN field as shown in Fig. 6. Simulation time is set to 864,000 s to imitate conditions during a whole day. In this scenario, we suppose that every RM is statically located and never powers off or leaves the WSN field and they participate in the WSN at the beginning of the simulation.

Figure 12 shows the CDF of the sensor sleep ratio. We can see that 64% sensors enjoy energy conservation and their sleep ratio becomes higher when we increase t_s from 20 ms to 100 ms. More specifically, 21 sensors (42%) sleep for more than half of the total simulation time when $t_s = 20$ ms and the number rises to 28 (56%) when $t_s = 100$ ms. The average sleep ratio over all sensors is 26.3% and 36.9% in the $t_s = 20$ ms and $t_s = 100$ ms cases, respectively. Figure 13 shows the CDF of packet end-to-end delay in such a scenario. When $t_s = 20$ ms, the average packet end-to-end delay increases 37.3% from 34.8 ms to 47.8 ms. In the $t_s = 100$ ms case, it increases by 164%, which is mainly caused by the buffering time in affected nodes. Fortunately, the average end-to-end delay of the lower 65% packets is 34.707 ms, almost the same as the average end-to-end delay when no RM exists.

Parameter t_s provides us design choices: in the delay-tolerance applications. We set larger t_s to obtain more energy saving. In the delay-sensitive case, a smaller t_s should be adopted to avoid significant delay raise.

2) Dynamic RM scenarios: Actually, RMs are dynamic and they may disappear from the WSN field at any time: exterior RMs (ubiquitously-used wireless devices) might move away or be powered off; interior ones (special sensors) may become normal when their power become scarce too. Take a laptop in our lab for example. It is powered on when its owner begins work at 8:12, 14:07 and 19:35 and when it is shut down after the owner leaves for meals (rests) at 11:46, 18:11 and 22:27. We use 14 laptops and 24-hour traces (can be found in our simulation source code) as input and run the simulation again. Results are shown in Figs. 14 and 15.

In a dynamic scenario, silent RMs may become active after the primary RM disappears. Hence, more sensors (86% as shown in Fig. 14) enjoy energy conservation. However, the sleep ratio of sensors becomes gentler due to the fact that few RMs would exist at night. The average sleep ratios over all sensors are 24.1% and 26.6% in $t_s = 20$ ms and $t_s = 100$ ms situations, respectively. Figure 13 shows the CDF of packet end-to-end delays. Average packet delay increases by 31.9% when $t_s = 20$ ms and 161% when $t_s = 100$ ms. Similar to the static RM scenario, most of the data (about 70%) just experience gentle delay raise while the remaining 30% experience significant delay raise due to buffering times in affected nodes.

Although only forward nodes could benefit from RMs, node types are variant. With dynamic appearance/disappearance of RMs, source nodes may become forward ones and enjoy energy saving. To improve the energy conservation efficiency,

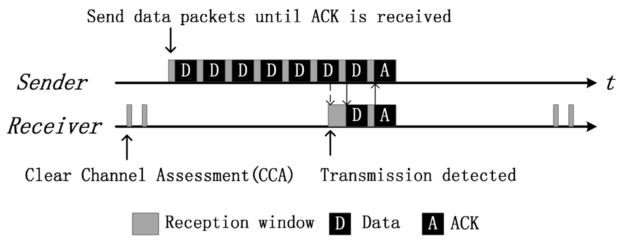

Send data packets until ACK is received
Sender
Receiver
D
A
t
Clear Channel Assessment(CCA)
Transmission detected
Reception window
D Data
A ACK

Fig. 16. Basic concept of ContikiMAC.

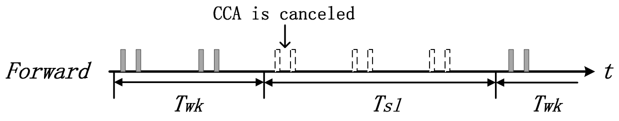


Fig. 17. Dual cyclings of forward nodes.

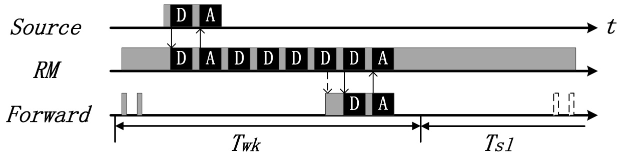


Fig. 18. Data forwarding process after applying ECARM in duty-cycled WSNs.

we could deploy specific RMs at night. We also suggest to distribute RMs in a uniform manner as only one active RM is allowed in its communication range.

V. Applying ECARM in duty-cycled WSNs

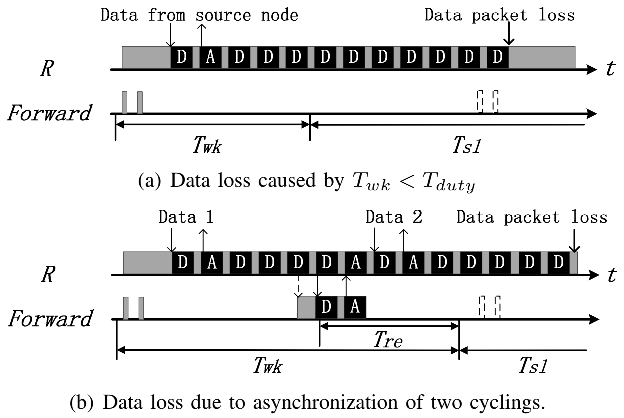
The above sections assume that every sensor keeps awake before RM participation. Here, we focus on applying ECARM in a WSN that is already duty-cycled under duty-cycling schemes such as ContikiMAC [19] and X-MAC [20]. The core concepts of these cycling methods are similar and we choose ContikiMAC as a paradigm.

A. Introduction of ContikiMAC

In order to save energy, WSNs are duty-cycled [31]. A widely-adopted duty-cycling protocol is ContikiMAC [19], the default radio duty-cycling mechanism in the Contiki operating system [32].

We show the basic concept of ContikiMAC [19] in Fig. 16. Sensors spend most of their lifetime in the sleep mode and periodically wake up to perform Clear Channel Assessment (CCA) to check for radio activity. Whenever a transmission event is detected, the receiver keeps awake to receive the next complete packet. If the packet is aimed at the receiver, it sends back a link layer acknowledgment (ACK). Otherwise, it drops the packet and returns to the sleep mode instantly.

To send a packet, a sender repeatedly transmits the same packet until an ACK is received. Without acknowledgement, the packet would be lost after a maximum transmission time of T_{duty}, which is the duty-cycling period of ContikiMAC. The radio check or the CCA period is also set to T_{duty} to ensure any transmission is reliably detected by two consecutive CCA groups. (Each group consists of two CCAs).



(a) Data loss caused by $T_{wk} < T_{duty}$

(b) Data loss due to asynchronization of two cyclings.

Fig. 19. Severe data loss may happen after applying ECARM in duty-cycled WSNs.

B. Analysis

Compared to ContikiMAC, ECARM conserves energy from a totally different perspective, i.e., opportunistically leveraging powerful RMs. We believe that applying ECARM in a duty-cycled WSN will generate better results than each alone. To avoid confusion, the terms wake-up/sleep cycling and duty-cycling are exclusively used by ECARM and ContikiMAC.

In a duty-cycled WSN, sensors keep duty-cycling all the time. ECARM changes sensor behaviors in the same way as described in Section II: Source nodes reelect RMs as their new next-hops while forward nodes experiencing dual cyclings, i.e., wake-up/sleep cycling and duty-cycling, at the same time. The two cycling methods are independent and compatible, which coexist on forward nodes as shown in Fig. 17. During T_{wk} time, forward nodes perform duty-cycling as if no RM exists during T_{sl}, nodes never wake up to check the channel.

Due to the existence of always-on RMs, time and energy required for source nodes to perform packet transmission have been significantly decreased. (See Fig. 18.) Meanwhile, under the principle of ECARM, whenever packets are received, RMs alternately relay them to a wake-up forward node and other forward nodes could stay in the sleep mode for energy saving.

Therefore, the energy conservation capacity of ECARM and ContikiMAC are both improved. As to ECARM, the beneficiaries of RMs expand from only forward nodes to all sensors within its coverage area. In terms of ContikiMAC, the energy conservation efficiency would be increased after the participation of RMs (verified in Subsection V-C).

Nevertheless, we indeed observe severe packet loss after ECARM application. We deem it might be caused by two factors. The first one is $T_{wk} < T_{duty}$. For example, as shown in Fig. 19(a), the second CCA group of forward nodes is canceled during T_{sl}, which means the transmission activity of RM R could not be detected. After R keeps transmission for T_{duty} time, the packet would have been lost. The second factor is denoted as T_{re}, i.e., we divide T_{wk} into two parts based on the last CCA group within T_{wk} and the latter part is T_{re} (shown in Fig. 19(b)). The transmission initiated during T_{re} (Data 2 in Fig. 19(b)) would be lost and the longer T_{re}, the higher loss rate. Fortunately, both factors could be alleviated by larger T_{wk}, i.e., larger time slot t_s.

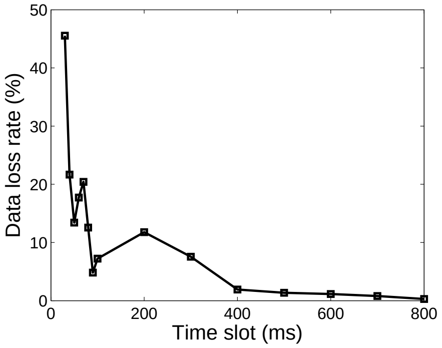


Fig. 20. Data loss rate vs. time slot t_s when check-rate is set to 20 HZ).

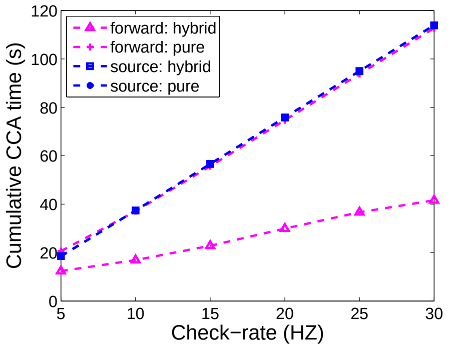


Fig. 21. Cumulative channel clear assessment time.

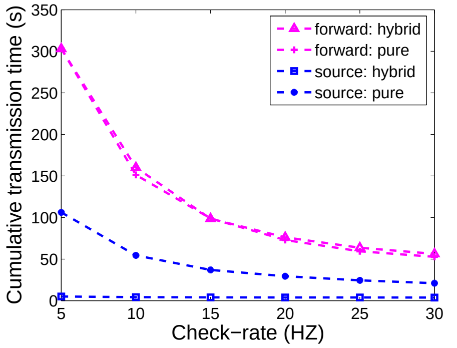


Fig. 22. Cumulative transmission time.

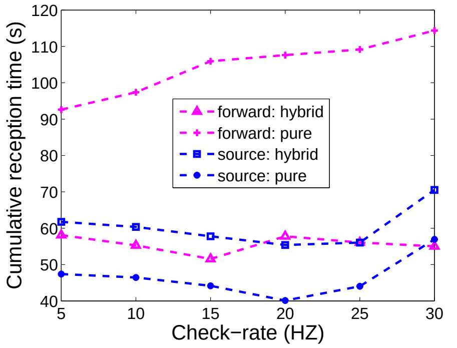


Fig. 23. Cumulative reception time.

C. Evaluation

We now verify our hybrid scheme through simulations, again with the settings of Table II. ContikiMAC is developed based on [19] (fast sleep is omitted due to the Poisson manner data generation we assume) and incorporated into our simulation as the Radio Duty Cycling (RDC) layer under MAC. The reception procedure of RMs remains unchanged and the transmission procedure is revised to accommodate ContikiMAC. A controlled distribution WSN in Fig. 1(b) is adopted and R is assumed to be static and always active. We set the simulation time to 10,000 s and the results are shown in Figs. 20-24.

We fix the ContikiMAC check-rate at 20 HZ ($T_{duty} = 50$ ms) and vary the time slot t_s of ECARM to obtain various T_{wk} values. As shown in Fig. 20, the data loss rate decreases with larger t_s. After $t_s > 10T_{duty}$, data loss rate falls below 1%. We believe the fluctuation is mainly caused by T_{re}, which is highly correlated with the periods of the two cyclings.

Next, we record the cumulative CCA, transmission and reception time (including data reception aimed at the receivers or data overhearing aimed at others) of source and forward nodes at various check-rates. To assure low data loss rate, we keep $t_s = 10T_{duty}$. In order to show the energy conservation improvement of the hybrid scheme, we compare the situation without RM (pure ContikiMAC) and with RM (hybrid). All of the reported results are averaged over two types of nodes (source and forward nodes).

After ECARM application, forward nodes do not check the channel or receive packets during T_{sl}, which results in decreasing cumulative CCA and reception time (see Figs. 21 and 23). Furthermore, with increased check-rate, the cumulative CCA and reception time is reduced more significantly due to the avoidance of frequent radio check and unnecessary data reception. However, the cumulative transmission time remains uninfluenced (see Fig. 22) since the work load of forward nodes is not reduced.

In terms of source nodes, the cumulative CCA time remains the same. As we mentioned above, in the pure ContikiMAC scenario, source nodes repeatedly send the same packet and more attempts are required when check-rate becomes lower, which results in longer transmission time. After RM participation, repeated packet sending becomes unnecessary and the cumulative transmission time becomes irrelevant to check-rate (see Fig. 21). We note that the transmission time of source nodes would be reduced greatly in circumstances where the ContikiMAC check-rate is very low or source nodes undertake heavy workload. However, the hybrid scheme does not assure improvements in all aspects, and it indeed deteriorates the cumulative reception time of source nodes (see Fig. 23). This is because all source nodes send data packets to the same RM in the hybrid scenario.

Finally, we record the total wake-up time of source and forward nodes. The results are shown in Fig. 24. With the increasing check-rate, the total active time of the hybrid scheme reduces more for forward nodes while less for source nodes. It provides us design space to choose different energy saving degrees for the two types of nodes. We also average the

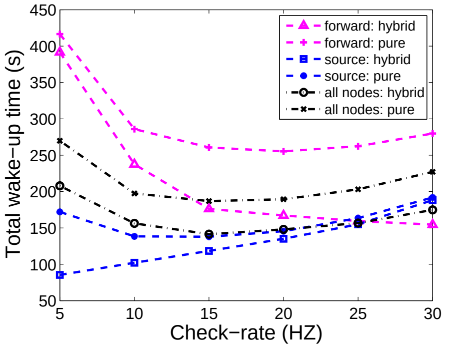


Fig. 24. Total wake-up time of source, forward and affected nodes.

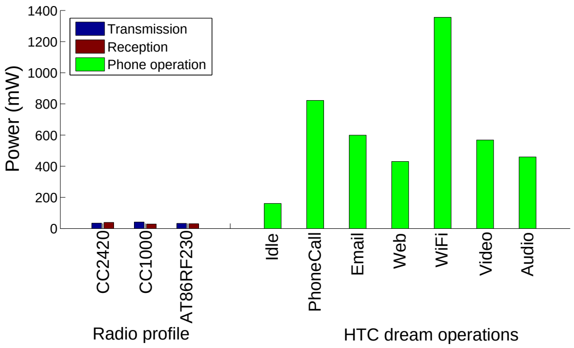


Fig. 25. Energy profile of HTC Dream.

total active time over all sensors in RM's coverage area. The total active time reduction rate varies from 20.9% to 24.3% at different check-rates, which is a significant improvement over pure ContikiMAC.

VI. Practical Issues

A. Security

After the adoption of ECARM, sensors might be open to public RMs and data security is weakened. Fortunately, a lot of approaches could be incorporated to ECARM to alleviate this problem. Here we just list several of them. First of all, it is unnecessary for source nodes to send all of their data to RMs. The knowledge of forward node's wake-up/sleep cycling is also possessed by source nodes. They can choose which data to be relayed by RMs and which data to be sent to wake up forward nodes directly. Secondly, authorization mechanisms such as [33] [34] [35] [36] [37] can be used to ensure that only entrusted RMs could participate in the sensornet. Moreover, cryptographic methods such as AES [38] and DES [39] can be performed by those entrusted powerful RMs to promote security.

We also recommend to deploy dedicated RMs for WSNs that require high security. As to WSNs devised for temperature/humidity monitoring, pollution detection, forest/mountain supervising etc., ECARM not only saves sensor's energy but also passes the valuable sensory information to individuals near the WSN, who really care about the environment.

B. Connectivity between RMs and sensors

Nowadays, the widespread wireless communication technologies equipped on daily-used mobile devices are Wi-Fi, bluetooth and 3G etc. However, most WSNs adopt the low power 802.15.4 standard as MAC/PHY protocol [30]. The communication gap between mobile devices and sensors exists because these technologies are incompatible. To deal with this problem, projects reported in [12] equip sensors with an extra bluetooth USB adapter. Therefore, a bluetooth communication link is established between RMs and sensors.

The main purpose of ECARM is to provide an general solution to save energy for WSNs with the assistance of RMs. ECARM is applicable whenever the communication among RMs and sensors becomes a reality, which can be obtained through 802.15.4 USB adapters or newly emerged 802.15.4 phones [40].

C. Data exchange

There exist three kinds of data exchange methods after RMs are included: sensor → RM → sink, sensor → RM → Internet and sensor → RM → sensor → sink. In the first way, RMs collect data from sensors and transmit them directly to the sink through technologies such as Wi-Fi. However, it assumes that RMs and sinks are closely located such that they can communicate with each other directly, which is only practical in small WSN scenarios. If the collected data are transmitted whenever RMs move close enough to the sink, it will definitely cause huge data delay since physical movement is far slower than wireless communication. Another reason we do not adopt Wi-Fi is its huge energy consumption, which will be discussed in Subsection VI-D. In the second way, sensing data are directly published to the Internet by RMs, which has two major drawbacks: it is undesirable to RMs that Internet access will cause data flow cost; it is unsuitable for scenarios where the sink is installed with synthesized control abilities, implying that a central controller is required to decide which data can be sent to the Internet and which data need to be stored privately. The third data exchange method avoids the above defects and therefore is adopted in this paper.

D. Energy Consumption of RMs

As we mentioned earlier, RMs either have direct power access or never run out of power with daily charging. However, ECARM would increase the energy consumption of RMs, which is undesirable for RM users. Fortunately, the energy consumption of an extra 802.15.4 compatible radio is truly limited. We list the transmission/reception power of several commonly-used sensor-radios including CC2420 [41], CC1000 [42] and AT86RF230 [43] shown in Fig. 25. Their energy-consumption is minimal even compared with most power-constrained RMs i.e., mobile phones. (The energy profile of HTC dream [44] is also shown in Fig. 25).

The holder of RMs will be glad to enable ECARM function if the sensory data in WSNs are user-interested. (Note that providing user interest information is the main purpose of most WSNs). Further, some incentives could be offered to promote user participation. For example, we can exchange the amount of sensory data passing through user's RMs with free phone traffic, which will attract lots of RMs.

VII. Related Work

Projects reported in [12] leverage the powerful connection ability of smart phones to provide sensors isolated from Internet access. Multiple practical applications in oil fields and offices have verified the feasibility and effectiveness of the idea to treat ubiquitously-used wireless devices as data mules. Meanwhile, Yunhao Liu et al. showed room for the improvement of the existing duty-cycling methods in [31]. The former projects motivate us to propose ECARM to resolve the energy-constraint problem of WSNs and the latter work shows us the potential benefit to combine duty-cycling with RMs.

There has been tremendous amount of work targeted at energy conservation in WSNs. Here we just discuss those most relevant to ECARM. For convenience, we classify them into three categories [8]: location-driven approaches, connectivity-based approaches and mobility-based approaches.

A. Location-driven Approach

Location-driven approaches [6] [45] make the presumption that every sensor knows the actual location of itself and the sink. With location information, GAF [45] divides the network field into small rectangle grids, and sensors in the same grid are equivalent in terms of routing function to sensors in other grids. At any time, in a certain grid, only one wake-up node is enough and others could go to sleep to save energy. Sensors in the grid of GAF [45] are similar to forward nodes in RM's coverage area of ECARM. However, in ECARM, the location-is-known assumption is removed and RMs are in charge of the coordination rather than nodes themselves. The coverage area of source nodes is split into a number of regions based on their distance from the sink in GeRaF [6]. Source nodes prefer wake-up nodes in regions closest to the sink to act as data relay. In ECARM, nodes are classified by their connection relationships rather than their distance from the sink and every forward node has the same priority.

B. Connectivity-based Approach

Connectivity-based approaches [46] [47] adaptively elect nodes with longer expected lifetime as coordinators and put others into the sleep mode. Most of these approaches include two parts: an algorithm that minimizes the number of coordinators and meanwhile guarantees the connectivity of the whole network; a protocol that exchanges power consumption information and negotiates the coordinator election process a-mong sensors. In ECARM, coordinators are resourceful mules rather than normal sensors. Thus, the complicated coordinator election algorithm and the burdensome power-surplus information exchange process is avoided. The energy conservation balance is obtained by the dynamic nature of RMs. Compared to connectivity-based approaches, one of the most obvious advantages of ECARM is simplicity.

C. Mobility-based Approach

Mobility-based approaches either assume that the sink (or sensors) is equipped with actuator [26] [48] or depend on mobile relays [13] [14] [15] [16] such as animals, vehicles, and robots. With the assistance of mobilizers, sensor's communication workload is relieved, i.e., the long range or multi-hop communication is replaced by short range or single-hop communication. Mobilizers receive data from sensors and relay them to the sink either through physical movement or multi-hop transmission. In this way, sensors spend most of their lifetime in the sleep mode and the energy is conserved. However, these methods either cause huge delay due to the limited physical speed of mobilizers or assume that sensors are aware of the instant location of the mobilizer. ECARM borrows the idea that takes advantage of mules, but it does not utilize their mobility. Sensors select out mules with long sojourn time within the WSN field and leverage their abundant power, computation, and connectivity resource for energy-saving. As a result, ECARM avoids the huge delay increase caused by most mobility-based approaches.

VIII. Conclusion and Future Work

In this paper, we presented ECARM, a novel mechanism that opportunistically utilizes resourceful mules to reduce power consumption in WSNs. We designed every part of ECARM and then simulated it on the OMNET++ platform. Through extensive simulations, we prove the correctness and energy conservation effectiveness of ECARM. Finally, we promote the efficiency of ContikiMAC by at least 20.9% after we apply ECARM in a WSN with ContikiMAC.

However, the application of ECARM in duty-cycled WSNs in this paper is crude in some aspects and the improvement is planned as our future work. Here we just list some of them: (1) The wake-up/sleep cycling of ECARM could be synchronized with the duty-cycling of ContikiMAC, which will reduce the data loss rate. (2) Incentive frameworks to promote deployment of ECARM considering that today's Internet is evolving to meet the continuous changing application requirements [49]. (3) Burst data transmission methods (e.g., [50]) could be adopted after RMs receive data from source nodes, so that the energy conservation efficiency will be further optimized.

Acknowledgment

The authors would like to thank the editor and anonymous reviewers for their constructive comments and suggestions. This research was supported in part by NSFC Project (61170292, 61472212 and 61161140454), National Science and Technology Major Project (2012ZX03005001), 973 Project of China (2012CB315803), 863 Project of China (2013AA013302) and EU MARIE CURIE ACTIONS EVANS (PIRSES-GA-2010-269323 and PIRSES-GA-2013-610524).

References

[1] G. Simon, M. Maróti, Á. Lédeczi, G. Balogh, B. Kusy, A. Nádas, G. Pap, J. Sallai, and K. Frampton, "Sensor network-based countersniper system," in *Proc. ACM SenSys*, 2004, pp. 1–12.

[2] J. Biagioni, T. Gerlich, T. Merrifield, and J. Eriksson, “Easytracker: automatic transit tracking, mapping, and arrival time prediction using smartphones,” in *Proc. ACM SenSys*, 2011, pp. 68–81.

[3] G. Tolle, J. Polastre, R. Szewczyk, D. Culler, N. Turner, K. Tu, S. Burgess, T. Dawson, P. Buonadonna, D. Gay *et al.*, “A macroscope in the redwoods,” in *Proc. ACM SenSys*, 2005, pp. 51–63.

[4] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava, “Optimizing sensor networks in the energy-latency-density design space,” *IEEE Trans. Mobile Computing*, vol. 1, no. 1, pp. 70–80, Jan. 2002.

[5] Z. Kong and E. M. Yeh, “Distributed energy management algorithm for large-scale wireless sensor networks,” in *Proc. ACM MobiHoc*, 2007, pp. 209–218.

[6] M. Zorzi and R. R. Rao, “Geographic random forwarding (geraf) for ad hoc and sensor networks: multihop performance,” *IEEE Trans. Mobile Computing*, vol. 2, no. 4, pp. 337–348, Oct. 2003.

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

[8] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, “Energy conservation in wireless sensor networks: A survey,” *Elsevier Ad Hoc Networks*, vol. 7, no. 3, pp. 537–568, May. 2009.

[9] C. Buratti, M. Martalo, G. Ferrari, and R. Verdone, *Sensor Networks with IEEE 802.15.4 Systems (Signals and Communication Technology)*. Springer-Verlag Berlin and Heidelberg GmbH, 2011.

[10] Y. Zhang, L. Sun, H. Song, and X. Cao, “Ubiquitous wsn for healthcare: Recent advances and future prospects,” *IEEE Internet of Things Journal*, vol. 1, no. 4, pp. 311–318, 2014.

[11] M. Erol-Kantarci and H. T. Mouftah, “Wireless sensor networks for cost-efficient residential energy management in the smart grid,” *IEEE Trans. Smart Grid*, vol. 2, no. 2, pp. 314–325, Jun. 2011.

[12] U. Park and J. Heidemann, “Data muling with mobile phones for sensornets,” in *Proc. ACM SenSys*, 2011, pp. 162–175.

[13] W. Wang, V. Srinivasan, and K.-C. Chua, “Using mobile relays to prolong the lifetime of wireless sensor networks,” in *Proc. ACM MobiCom*, 2005, pp. 270–283.

[14] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, “Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet,” in *ACM Sigplan Notices*, vol. 37, no. 10, 2002, pp. 96–107.

[15] A. A. Somasundara, A. Kansal, D. D. Jea, D. Estrin, and M. B. Srivastava, “Controllably mobile infrastructure for low energy embedded networks,” *IEEE Trans. Mobile Computing*, vol. 5, no. 8, pp. 958–973, 2006.

[16] S. Jain, R. C. Shah, W. Brunette, G. Borriello, and S. Roy, “Exploiting mobility for energy efficient data collection in wireless sensor networks,” *Springer MONET*, vol. 11, no. 3, pp. 327–339, Jun. 2006.

[17] K. Chebrolu, B. Raman, N. Mishra, P. K. Valiveti, and R. Kumar, “Brimon: a sensor network system for railway bridge monitoring,” in *Proc. ACM Mobisys*, 2008, pp. 2–14.

[18] T. Shu and M. Krunz, “Coverage-time optimization for clustered wireless sensor networks: a power-balancing approach,” *IEEE/ACM Trans. Networking*, vol. 18, no. 1, pp. 202–215, 2010.

[19] A. Dunkels, “The contikimac radio duty cycling protocol,” Swedish Institute of Computer Science, Tech. Rep., 2011.

[20] M. Buettner, G. V. Yee, E. Anderson, and R. Han, “X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks,” in *Proc. ACM SenSys*, 2006, pp. 307–320.

[21] X. Zhang and K. G. Shin, “E-mili: energy-minimizing idle listening in wireless networks,” *IEEE Trans. Mobile Computing*, vol. 11, no. 9, pp. 1441–1454, Sept. 2012.

[22] *AR5213 Preliminary Datasheet*, Atheros Comm, 2004.

[23] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, “Collection tree protocol,” in *Proc. ACM SenSys*, 2009, pp. 1–14.

[24] D. C. Dhanapala, A. P. Jayasumana, and Q. Han, “On random routing in wireless sensor grids: A mathematical model for rendezvous probability and performance optimization,” *Journal of Parallel and Distributed Computing*, vol. 71, no. 3, pp. 369–380, 2011.

[25] D. O. Olguín, B. N. Waber, T. Kim, A. Mohan, K. Ara, and A. Pentland, “Sensible organizations: Technology and methodology for automatically measuring organizational behavior,” *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Trans.*, vol. 39, no. 1, pp. 43–55, 2009.

[26] J. Luo and J.-P. Hubaux, “Joint mobility and routing for lifetime elongation in wireless sensor networks,” in *Proc. IEEE INFOCOM*, 2005.

[27] M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi, “The flooding time synchronization protocol,” in *Proc. ACM SenSys*, 2004, pp. 39–49.

[28] S. Ganeriwal, R. Kumar, and M. B. Srivastava, “Timing-sync protocol for sensor networks,” in *Proc. ACM SenSys*, 2003, pp. 138–149.

[29] A. Varga *et al.*, “The omnet++ discrete event simulation system,” in *Proc. EUROSIS ESM*, vol. 9, 2001.

[30] *Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*, IEEE 802.15.4-2006 Std., 2006.

[31] Z. Cao, Y. He, and Y. Liu, “L2: Lazy forwarding in low duty cycle wireless sensor networks,” in *Proc. IEEE INFOCOM*, 2012, pp. 1323–1331.

[32] A. Dunkels, B. Gronvall, and T. Voigt, “Contiki-a lightweight and flexible operating system for tiny networked sensors,” in *Proc. IEEE LCN*, 2004, pp. 455–462.

[33] D. Liu and P. Ning, “Efficient distribution of key chain commitments for broadcast authentication in distributed sensor networks,” in *Proc. Internet Society NDSS*, vol. 3, 2003, pp. 263–276.

[34] M. Bohge and W. Trappe, “An authentication framework for hierarchical ad hoc sensor networks,” in *Proc. ACM WiSec*, 2003, pp. 79–87.

[35] C. Hennebert and J. Dos Santos, “Security protocols and privacy issues into 6lowpan stack: A synthesis,” *IEEE Internet of Things Journal*, vol. 1, no. 5, pp. 384–398, 2014.

[36] S. Keoh, S. Kumar, and H. Tschofenig, “Securing the internet of things: A standardization perspective,” *IEEE Internet of Things Journal*, vol. 1, no. 3, pp. 265–275, 2014.

[37] W. Tan, K. Xu, and D. Wang, “An anti-tracking source-location privacy protection protocol in wsns based on path extension,” *IEEE Internet of Things Journal*, vol. 1, no. 5, pp. 461–471, 2014.

[38] *Specification for the Advanced Encryption Standard (AES)*, National Institute of Standards and Technology Std., 2001. [Online]. Available: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[39] *Data Encryption Standard (DES)*, National Institute of Standards and Technology Std., 2001. [Online]. Available: http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf

[40] “Enabling zigbee and 802.15.4 in pda and mobile phones,” wireless sensor networks research group, 2009. [Online]. Available: http://www.sensor-networks.org/index.php?page=0902602615

[41] *CC2420 Datasheet*, Texas Instruments, 2007.

[42] *CC1000 Datasheet*, Texas Instruments, 2007.

[43] *IEEE 802.15.4 Transceiver AT86RF230 Datasheet*, Atmel, 2006.

[44] A. Carroll and G. Heiser, “An analysis of power consumption in a smartphone,” in *Proc. USENIX ATC*, 2010, pp. 21–21.

[45] Y. Xu, J. Heidemann, and D. Estrin, “Geography-informed energy conservation for ad hoc routing,” in *Proc. ACM MobiCom*, 2001, pp. 70–84.

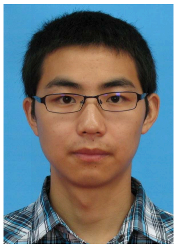
[46] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, “Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks,” in *ACM MobiCom*, 2001, pp. 85–96.

[47] O. Dousse, P. Mannersalo, and P. Thiran, “Latency of wireless sensor networks with uncoordinated power saving mechanisms,” in *Proc. ACM MobiHoc*, 2004, pp. 109–120.

[48] S. Basagni, A. Carosi, E. Melachrinoudis, C. Petrioli, and Z. M. Wang, “Controlled sink mobility for prolonging wireless sensor networks lifetime,” *Wireless Networks*, vol. 14, no. 6, pp. 831–858, 2008.

[49] X. Ke, Z. Min, H. GuangWu, Z. Liang, Z. YiFeng, L. Ying, W. JianPing, and W. Ning, “Towards evolvable internet architecture-design constraints and models analysis,” *Science China*, 2014.

[50] S. Duquennoy, F. Österlind, and A. Dunkels, “Lossy links, low power, high throughput,” in *Proc. ACM SenSys*, 2011, pp. 12–25.



Yi Qu received the BEng degree in software engineering from University of Electronic Science and Technology of China in 2011. Currently he is a Ph.D. student in the Department of Computer Science & Technology of Tsinghua University, Beijing, China. His research interests include wireless network and wireless sensor network.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2014.2371056, IEEE Internet of Things Journal

Ke Xu (M'02-SM'09) received his Ph.D. from the Department of Computer Science & Technology of Tsinghua University, Beijing, China, where he serves as a full professor. He has published more than 100 technical papers and holds 20 patents in the research areas of next generation Internet, P2P systems, Internet of Things (IoT), network virtualization and optimization. As a member of ACM and an associate editor of Springer journal Networking Science, he has guest edited several special issues in IEEE and Springer Journals. He is serving as an associate editor of Springer journal Networking Science. He is also holding visiting professor position of University of Essex.

Jiangchuan Liu (S'01-M'03-SM'08) received the BEng degree (cum laude) from Tsinghua University, Beijing, China, in 1999, and the Ph.D. degree from The Hong Kong University of Science and Technology in 2003, both in computer science. He was a recipient of Microsoft Research Fellowship (2000), a recipient of Hong Kong Young Scientist Award (2003), and a co-inventor of one European patent and two US patents. He co-authored the Best Student Paper of IWQoS'08 and the Best Paper (2009) of IEEE Multimedia Communications Technical Committee (MMTC). He is currently an associate professor at the School of Computing Science, Simon Fraser University, British Columbia, Canada, and was an assistant professor in the Department of Computer Science and Engineering at the Chinese University of Hong Kong from 2003 to 2004. His research interests include multimedia systems and networks, wireless ad hoc and sensor networks, and peer-to-peer and overlay networks. He is an associate editor of IEEE Transactions on Multimedia, and an editor of IEEE Communications Surveys and Tutorials. He is a senior member of IEEE and a member of Sigma Xi.

Wenlong Chen born in 1976, Ph.D. He is currently a lecturer at the college of Information Engineering of Capital Normal University. His research interests include network protocol and network architecture.