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/TII.2015.2412093, IEEE Transactions on Industrial Informatics

COPE: Improving Energy Efficiency with Coded Preambles in Low Power Sensor Networks

Yi Gao, *Member, IEEE,* Wei Dong, *Member, IEEE,* Lizhi Deng, Chun Chen, *Member, IEEE,* Jiajun Bu, *Member, IEEE*

***Abstract*—Energy efficiency is one of the most important factors that affect the applicability of Wireless Sensor Networks (WSNs) in many practical scenarios. Many low power MAC protocols have been proposed in the past decade to improve the energy efficiency of radio operations which cost most of the energy of sensor nodes. In these low power MAC protocols, preambles are widely used to wake up the receivers asynchronizely. However, the data delivery potential of these preambles has not been exploited. In this paper, we propose COPE, which exploits the data delivery potential of preambles by encoding the preambles by network coding. COPE has two salient features. First, a passive receiver set selection scheme enables nodes to decide whether to receive the overheard preamble packets, without introducing extra communication overhead. Second, COPE supports multiple routing primitives, such as unicast and broadcast, making it be a versatile 2.5 layer between the low power link layer (layer 2) and the network layer (layer 3). We analyze COPE by a novel analytical model. Results show that COPE is able to improve the energy efficiency of both unicast and broadcast significantly. We also implement COPE in TinyOS/TelosB platform and evaluate its energy efficiency. Results show that COPE reduces the radio-on-time significantly in practical network settings.**

***Index Terms*—Energy efficiency, sensor network, network coding.**

I. Introduction

Energy efficiency has been one of the most important factors that affect the applicability of sensor networks in many practical scenarios [1], [2]. Since radio operations (i.e., transmitting, receiving and idle listening) cost most of the energy [3], many low power protocols have been proposed in the past decade [4]–[11]. By turning off the radio most of the time (i.e., low duty cycle), these protocols successfully reduce the energy consumption and prolong the lifetime of the network significantly.

Due to the wireless dynamics, asynchronized low power approaches, which do not require time synchronization, have been shown to be more appropriate in real deployments [12]. In asynchronized low power approaches, a sensor node does not know when its receivers will wake up and be able to receive packets. Therefore, the sender usually needs to transmit a "preamble" to wake up its targeted receivers [6], [7], [9]. BMAC [6] is the first asynchronized low power MAC protocol for sensor networks. All nodes wake up periodically with a predetermined interval. When a node needs to send data, it transmits a fixed length preamble, which is longer than the wake up interval, before sending the actual data. Then the receivers wake up during that preamble and receive the data. XMAC [7] uses short packets with targeted destination to be the preamble. When a receiver wakes up and receive a short packet whose destination is the receiver, it sends back an ACK packet to the sender. Then the sender can stop sending the preamble packets and start sending actual data packets. XMAC reduces the length of preamble significantly compared with BMAC. As the default low power protocol in TinyOS, BoXMAC replaces the short packets in XMAC with actual data packets. Then after the targeted receiver ACKs the preamble packet, the sender can stop sending immediately. BoXMAC further reduces one packet transmission compared with XMAC. However, each receiver using BoXMAC can only receive at most one unique packet from the preamble packets. All other overheard preamble packets are discarded.

Although these low power approaches have successfully reduced the length of the preambles, the preambles are still only used to wake up the targeted receivers. When the wake up interval is 512 ms, the sender needs to transmit the preamble packets for 256 ms (25∼50 preamble packets) on average, causing high energy consumption. Considering the high energy consumption of preamble packets transmission and the constrained channel resource, using preamble packets only to wake up receivers has become the fundamental inefficiency in terms of both energy and channel resource.

In order to exploit the data delivery potential of preambles, we propose COPE, a COded PrEamble design for low power wireless sensor networks. Intuitively, at the sender side, COPE encodes multiple native packets and transmits the encoded packets as preamble packets to wake up one or multiple receivers. Then at the receivers' side, COPE decodes the encoded packets and obtains the native packets. However, due to the ad hoc nature of sensor nodes and the wireless dynamics, there are two practical challenges. First, it is challenging to select receivers to receive the encoded preamble packets. Since a node should keep awake and be in high energy state when it is receiving packets, the receiver selection should be conducted carefully not to

Y. Gao, W. Dong, L. Deng, C. Chen and J. Bu are with the College of Computer Science, and the Zhejiang Provincial Key Laboratory of Service Robot, Zhejiang University, Zhejiang, China. Corresponding author: Wei Dong. Email: dongw@zju.edu.cn.

We would like to thank the editors and the anonymous reviewers for their insightful comments. This work was supported by the National Science Foundation of China under Grants No. 61202402, the Fundamental Research Funds for the Central Universities, National Key Technology R&D Program under Grant 2012BAI34B01, and Demonstration of Digital Medical Service and Technology in Destined Region.

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/TII.2015.2412093, IEEE Transactions on Industrial Informatics

cause unnecessary energy consumption. Further, given the ad hoc nature of sensor nodes, how to select receivers without introducing extra communication is the key challenge in the design of COPE. Second, since there are multiple routing primitives (e.g., unicast, multicast and broadcast) in real sensor network deployments, it is challenging to support these routing primitives at the same time. In order to address these two challenges, COPE includes a *passive receiver set selection* design to select the receivers for all three communication primitives. COPE has two salient features. First, it does not require extra communication among sensor nodes to achieve efficient receiver selection, improving the energy efficiency significantly. Second, it is a versatile 2.5 layer between the low power link layer (layer 2 [9]) and the network layer (layer 3 [13], [14]). COPE supports multiple routing primitives, such as unicast, multicast and broadcast.

We theoretically analyze the energy efficiency of COPE by an novel analytical model. Results show that COPE is able to achieve higher energy efficiency compared with a traditional approach. We also implement COPE in TinyOS and evaluate its performance in various network settings. Results show that COPE improves the energy efficiency significantly, compared with a traditional preamble based approach. The main contributions of this paper are as follows.

- For the first time, we exploit the data delivery potential of preamble packets in asynchronized low power sensor networks.
- We propose COPE, a coded preamble design for improving the energy efficiency of low power sensor networks. COPE has two salient features. First, it does not introduce extra communication, improving energy efficiency significantly. Second, it is a versatile 2.5 layer, which support all three common routing primitives.
- We implement COPE and evaluate its performance in various network settings. We also propose a novel analytical model to analyze its performance. Results show that COPE achieves much higher energy efficiency compared with a traditional preamble based low power approach.

The rest of the paper is organized as follows: Section II discusses the related work and Section III gives some background information of the network coding technique used in COPE. Section IV presents the design of COPE. Section V analyzes the energy efficiency of COPE by a novel analytical model. Section VI presents the evaluation results and Section VII concludes this work.

II. Related Work

The related work of COPE include the low power MAC protocols, the low power routing protocols and network coding techniques in sensor networks.

A. Low Power MAC Protocols in Sensor Networks

Low power MAC protocols have been widely used in many sensor networks [15]–[17] to improve energy efficiency. During the past years, many low power MAC protocols have been proposed to improve the energy efficiency of sensor networks. We only discuss some representative approaches. When time synchronization (local or global) is available, synchronized MAC protocols can schedule data transmission to achieve energy efficiency. SMAC [5] is the first synchronized MAC protocol for sensor networks. It uses fixed active/sleep intervals for all sensor nodes and reduces the radio-on-time by 50%. Since time synchronization usually needs non-negligible overhead, many asynchronized MAC protocols are proposed. As mentioned in the introduction section, XMAC [7] and BoXMAC [9] are two typical asynchronized MAC protocols. These asynchronized MAC protocols need to transmit a preamble to wake up the targeted receivers before data transmission. Different with these works, COPE exploits the data delivery potential of the preamble packets to improve the energy efficiency.

There are also another research direction of asynchronized MAC protocols, which are receiver-initiated protocols [11], [18]. Instead of using preambles sent by senders, receiver-initiated protocols let the nodes send "probes" periodically. Then a sender can start sending data after it receives a probe from its targeted receiver. The performance gain of receiver-initiated MAC protocols compared with sender-initiated MAC protocols depends on the network settings, such as the data rate and network density.

B. Low Power Routing Protocols in Sensor Networks

Besides low power MAC protocols, low power routing protocols [19]–[23] are also important to achieve energy efficiency. Several protocols [19] assume that a node knows the wake up schedules of its neighboring nodes. Based on these wake up schedules, these approaches propose different routing methods to reduce the data forwarding delay, or improve the data forwarding robustness in case of link failure. Different with these protocols, COPE does not assume local wake up information which can introduce non-negligible overhead in dynamic networks. There are also routing protocols [21], [22] focusing on balancing the energy consumption across the network. FAF-EBRM [21] is an energy-balanced routing method based on forward-aware factor. EARQ [22] provides real-time, reliable delivery of a packet, while considering energy awareness. These approaches are orthogonal to COPE proposed in this paper.

C. Network Coding in Sensor Networks

Network coding techniques have been widely used to improve the performance of sensor networks. In order to be loss resilience, several data dissemination protocols use network coding techniques to reduce the number of retransmission packets [24]–[26]. These code-based protocols differ from each other in the coding techniques being used. For example, Rateless Deluge [24] employs random linear code, and ReXOR [26] employs XOR-based code. These approaches all assume an always on link layer of the sensor network. Different from these approaches, COPE aims at using network coding to improve the energy efficiency over a low power link layer. DutyCode [27] uses network coding for energy efficient data flooding in low power sensor networks. The basic

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/TII.2015.2412093, IEEE Transactions on Industrial Informatics

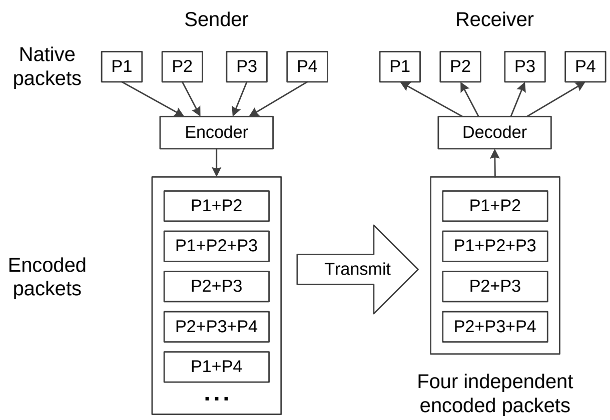


Fig. 1. An example of using fountain code to transmit data packets. Linear code is used in this example.

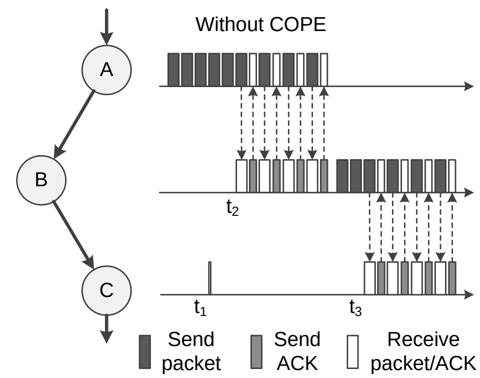


Fig. 2. The low power unicast communication without COPE to deliver four data packets. Node B wakes up at t_2 and starts receive data packets. Node C wakes up at t_1 and t_3, and it starts receiving packets from B at t_3.

idea of DutyCode is to exploit the redundancy in flooding applications that use network coding, and put a node to sleep state when a redundant transmission takes place. Different with DutyCode which focuses on data flooding, COPE supports multiple routing primitives including unicast, multicast and broadcast.

III. Background of Fountain Codes

In this section, we give a brief background of network coding. Specifically, we describe Fountain Codes [28] used in COPE. Foutain Codes are near optimal rateless codes for erasure channels. At the sender side, it encodes a number of native packets into encoded packets. At the receiver side, it is not important which encoded packets are lost during the transmission, what matters is how many encoded packets are correctly received. When sufficient encoded packets are received by the receiver, they can be decoded to native packets by the receiver. Figure 1 shows an example of using linear fountain code to transmit data packets in sensor networks. Suppose there are four native packets to be transmit at the sender side: P1, P2, P3 and P4. The encoder encodes these four native packets into a number of encoded packets. Then the sender transmits these encoded packets to the receiver. When the receiver receives four linear independent encoded packets, it decodes the received encoded packets and obtains the four native packets. In this example, the first four encoded packets are linearly independent. Therefore, the receiver can decode the first four encoded packets and obtain the four native packets.

IV. Design

This section presents the design of COPE, a coded preamble design for energy efficient data delivery in low power sensor networks. The basic COPE design requires a sender to first encode a number of native packets in its sending queue, then transmits the encoded packets as preamble packets to its receivers. At the receiver side, COPE decodes sufficient encoded packets and obtains the native packets. Since all nodes are in low duty cycle states, keeping awake and receiving preamble packets will consume extra energy. In order to improve the energy efficiency, a key design consideration of COPE is to determine which nodes should receive the preamble packets, i.e., *receiver set selection*. Specifically, COPE uses a passive receiver set selection approach (i.e., no extra communication overhead) to let the nodes locally determine whether to receive the preamble packets. In the following, we will describe the detailed design of COPE in unicast, multicast and broadcast.

A. Unicast and Multicast Communication

In sensor networks, unicast communication is widely used in one to one data delivery or many to one data collection [29]. Figure 2 shows a typical unicast scenario. Three nodes A, B and C are in the data delivery path. Packets transmitted to node A are forwarded to node B and further forwarded to node C, and then are forwarded to other nodes. We refer to the previous hop as the "child" of the current hop, and the next hop as the "parent" of the current. For example, node B is the parent of node A and the child of node C. Figure 2 shows the case without using COPE to deliver four data packets. When node A wants to send the four data packets, it continuously sends the first packet to wake node B. Then node B wakes up at time t_2 and receives the first data packet. After receiving the ACK packet from node B, node A sends the rest three data packets to B. Then node B forwards these four data packets to node C by a similar procedure.

Using COPE to deliver these four data packets differs the above approach as follows. First, COPE encodes the data packets and transmit them as encoded preamble packets. A receiver acknowledges the sender when it has received sufficient number of encoded preamble packets, instead of acknowledging every data packet. Therefore, the number of acknowledges are reduced. Second, COPE uses opportunistic routing technique to further improve the data delivery efficiency. In the previous example, node A will not only transmit data to its receiver B, but can also transmit data to node C opportunistically. We use an abstraction called "receiver set" to describe how to achieve proper addressing in the design of COPE. The receiver set of a sender includes two nodes, one is the parent p of the sender, and the other is the parent of p. The nodes in the receiver set should acknowledge the

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/TII.2015.2412093, IEEE Transactions on Industrial Informatics

packets sent from the sender. COPE uses a completely passive approach to coordinate the sender and receivers, so that there no extra communication overhead. For the sender (i.e., A), it does not need to know its receiver set. For the receiver (i.e., B), it can easily know that it is in the receiver set by checking the unicast destination field of the packet. For the parent of the receiver (i.e., C), it is also in the receiver set, because the destination (i.e., node B) of the packet is its child, which means that packets transmitted to the receiver will be further forwarded to it. In order to achieve this, each node keeps a set of child nodes. Whenever it receives a packet, it scans the set to determine whether it is in the receiver set of that packet.

In the example shown in Figure 3, the *receiver set* is $\{B, C\}$. Node C is in the receiver set, because the destination of the packets from node A (i.e., node B) is the child of node C. In fact, the link AC is a short cut of links AB and BC. This feature of *passive receiver set selection* avoids extra communication overhead. For other nodes that overhear the packets, they know that B is not their child node. When a node wakes up and receives a packet, it will first check whether it is in the receiver set. If the node is not in the receiver set, it will drop the packet and go back to sleep state. If C is not able to hear the packets from A, B becomes the only receiver in the receiver set. We then focus on the case when both node B and C can hear node A. Node A sends the coded preamble packets to the receiver set till it receives an ACK packet or reaches the maximum preamble time T_p. The detailed description of determining T_p is given in the next subsection.

There are two difference cases which are both shown in Figure 3.

Case 1: when node C receives sufficient encoded packets to decode before node B. It is usually because that C wakes up earlier then B. Note that in some extreme cases, although B wakes up earlier than C, it drops more encoded packets than C. Then node C will still receive sufficient encoded packets before node B. In Case 1, node C receives sufficient encoded packets (e.g., four in this example) and sends an ACK packet to the sender A. After node A has received the ACK packet, it stops sending encoded packets. The data delivery among these three nodes is completed since the last node C has received sufficient encoded packet, and it does not matter whether node B receives the packets. Compare this Case 1 with the case shown in Figure 2, COPE significantly reduces the energy consumption during the data delivery.

Case 2: when node B receives sufficient encoded packets to decode before node C. In this case, node C has not received sufficient encoded packets. Therefore, node B should forward the encoded packets to node C. In this example, node C has received two encoded packets from A, so it only needs two more encoded packets from node B to decode them. Compared with the case without COPE, COPE improves the energy efficiency significantly in case 2.

Compared with the traditional low power approach shown in Figure 2, COPE has two key differences. First, the preambles are not the repeated native packet, but the encoded packets of a number of native packets. When the receiver wakes up, it only sends ACK after it has received sufficiently encoded packets, instead of sending one ACK for each packet. Therefore, the number of ACKs is significantly reduced. Second, COPE uses a passive receiver set selection approach to enable data delivery from a short cut link, speeding up the data delivery process and improving the energy efficiency. Note that the performance gain of COPE mainly comes from the coded preamble design, the above opportunistic routing design further improves the data delivery efficiency of unicast in COPE.

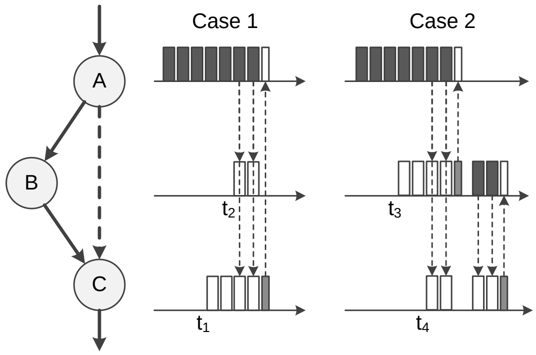


Fig. 3. COPE in unicast communication scenario where four native packets need to be delivered. The receiver set is $\{B, C\}$. Two difference cases are shown in the figure. t_1,t_2,t_3 and t_4 are the four times when node B and C wake up.

Multicast can be viewed as multiple simultaneously unicasts. Therefore, the multicast design of COPE is similar with the unicast, except the following differences. In unicast, a sender will stop sending encoded packets when it receives one ACK packet. In multicast, however, the sender should stop sending encoded packet when multiple ACKs is received. The number of ACKs are the number of targeted receivers.

B. Broadcast Communication

Broadcast communication is also widely used in low power sensor network for various purposes, such as neighbor discovery, data dissemination and link estimation. Without using COPE, broadcasting a number of packets over low power MAC works as follows. The sender keeps sending the first packet repeatedly for a predetermined time interval which is larger than the wake up interval T_w of the receivers. The receivers wake up and receive the first packet. Then the sender uses the same approach to send the rest of packets. If reliable transmission is required (e.g., code dissemination [24]), each receiver should send an NACK (which includes the information of the lost packets) to the sender. Then the sender retransmits those lost packets. Figure 4 shows this procedure.

Different from the above approach, broadcast of COPE is more energy efficient. When a sender wants to broadcast a number of packets, it encodes them and broadcast the encoded packets as preamble packets for a predetermined time T_p. T_p can be calculated by the following equation.

$$T_p = T_w + \alpha n T_e, \tag{1}$$

where T_w is the wake up interval of the receivers, n is the number of native packets encoded, T_e is the time of

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/TII.2015.2412093, IEEE Transactions on Industrial Informatics

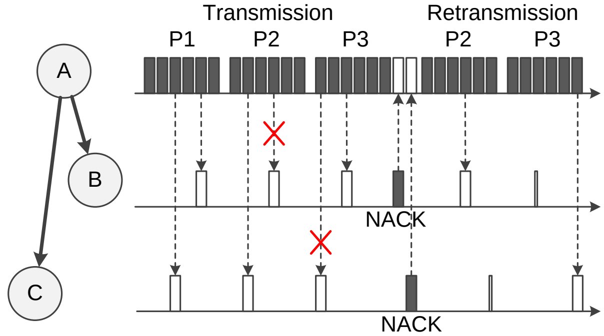


Fig. 4. Using low power broadcast communication without COPE to broadcast three data packets: P1, P2 and P3.

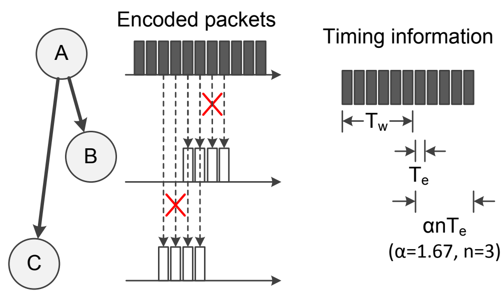


Fig. 5. COPE in low power broadcast communication scenario where three native packets need to be broadcasted. When the two receivers receive three linearly independent encoded packets, they are able to decode them and obtain the original three native packets.

transmitting one encoded packet and α is a constant greater than 1. Figure 5 shows an example where three packets are sent from the sender A and received by node B and C. There are two key differences compared with the case shown in Figure 4. First, the coded preamble design of COPE sends the three native packets by encoded packets in one receiver wake up interval, avoiding multiple rounds of packet broadcast. Second, the broadcast time T_p in COPE is longer than the wake up interval T_w of the receivers. The reason is that the receivers should receive sufficient number of encoded packets to decode them. If a receiver wakes up T_w later than the time when the sender starts sending packets, it will take nT_e to receive n encoded packets. Therefore, T_p should at least be $T_w + nT_e$. Considering packet losses and linearly dependent encoded packets, COPE uses a constant α to improve the robustness. Clearly, increasing α will increase the probability that the receivers successfully receive sufficient encoded packets, but also increase the energy consumption. We evaluate the impact of α in the evaluation section. As a result, the energy consumption of broadcasting the three packets is significantly reduced by COPE.

V. Analysis

In this section, we theoretically analyze the energy efficiency of COPE in both unicast and broadcast. Since radio operations cost most of the energy of sensor nodes [3], we use the *radio-on-time* as the metric of energy efficiency. We first give the following notations used in the analysis.

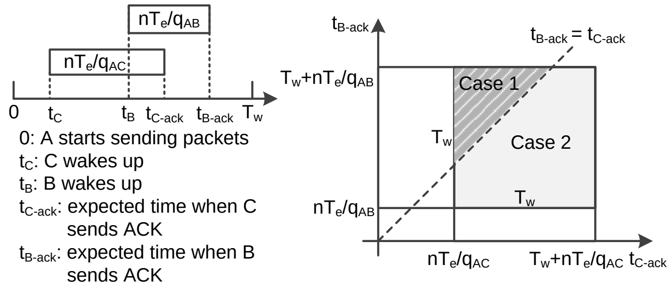


Fig. 6. Calculating the probability of case 1. It is the ratio of the triangle in the T_w^2 square.

- T_w: the wake up interval of the sensor node. We assume all nodes have the same wake up interval, which is a common low power setting in deployed networks [1].
- n: number of native packets need to be encoded and then be transmitted by either unicast or broadcast.
- T_e: the time to transmit a encoded packet.
- T_{dar}: the delay before a node enters sleep state after its previous packet receiving (i.e., delay after receiving).
- q_{XY}: the link quality of link XY. When it equals to 1, all packets transmitted over it can be corrected received.
- T_{rot}^X: the radio-on-time of node X during the data delivery.
- T_{cca}: the time to perform a Clear Channel Assessment (CCA). In low power sensor networks, a sensor node periodically wakes up and checks whether there is packet transmission near it. It is a constant about 11 ms in a typical sensor node [3].
- T_{csma}: the CSMA backoff time before a node actually transmits the packets. The initial CSMA backoff is 0∼9.8 ms and the congestion backoff is 0∼5 ms. We use the median value of the initial backoff(i.e., 4.9 ms) in the model.

Since the time to transmit an ACK packet is only about 0.2 ms, we omit this ACK time in the model. We also assume the encoded packets are linearly independent.

A. Unicast

Consider three nodes A, B and C that on the delivery path of n native packets. We calculate the total radio-on-time of these three nodes, starting from node A starts to send the packets and ending when node C receives all packets. Without using COPE, the expected total radio-on-time can be calculated as follows.

$$\begin{aligned}
E(T_{rot}^A) &= \frac{1}{2}T_w + \frac{nT_e}{q_{AB}} + T_{csma}, \\
E(T_{rot}^B) &= \frac{1}{2}T_w + \frac{nT_e}{q_{AB}} + \frac{nT_e}{q_{BC}} + T_{cca} + T_{csma}, \\
E(T_{rot}^C) &= \frac{nT_e}{q_{BC}} + T_{cca} + T_{dar}, \\
E(T_{rot}) &= E(T_{rot}^A) + E(T_{rot}^B) + E(T_{rot}^C), \\
&= T_w + 2 * nT_e(\frac{1}{q_{AB}} + \frac{1}{q_{BC}}) + 2T_{csma} + 2T_{cca} + T_{dar}.
\end{aligned} \tag{2}$$

Then we consider using COPE to delivery these n packets. Since there are two difference cases of unicast using COPE,

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/TII.2015.2412093, IEEE Transactions on Industrial Informatics

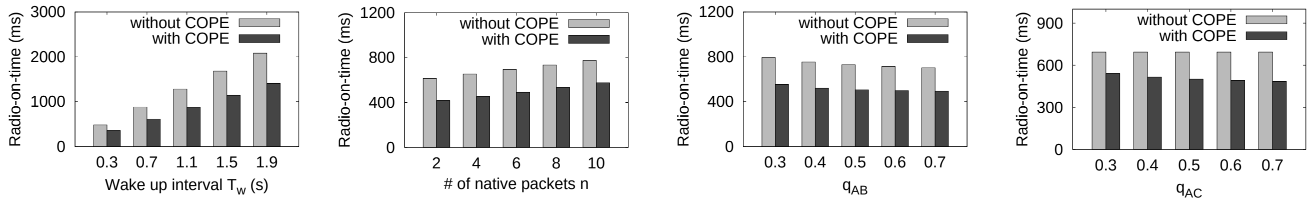


Fig. 7. Analytical results of unicast. This figure shows the impacts of the wake up interval T_w, the number of native packets n, q_{AB} and q_{AC}.

we first calculate the probability of each case. Without loss of generality, we assume the link quality of link AB is better than AC, i.e., $q_{AB} > q_{AC}$. Then the time that node B and C need to receive n encoded packets is nT_e/q_{AB} and nT_e/q_{AC}, respectively. Figure 6 shows the calculation of the probability of case 1. The wake up times of node B and C can be anytime between 0 and T_w, where node A starts sending packets at time 0. Then the T_w^2 square is the feasible region of the times these two nodes finish receiving. Then the probability of Case 1 (i.e., $t_{C-ack} < t_{B-ack}$) can be calculated as the ratio of the triangle in the T_w^2 square.

$$P(\text{Case-1}) = \frac{(T_w + nT_e/q_{AB} - nT_e/q_{AC})^2}{2T_w^2}. \tag{3}$$

Note that the above calculation assumes that at least one node in B and C receives sufficient encoded packets before node A stops sending packets. Otherwise the packets are lost. In this case, node A may retransmits these packets, depending on whether reliable transmission is required. The probability of Case 2 can be calculated as $1 - P(\text{Case-1})$. Then we calculate the expected radio-on-time of each case.

Case 1. The expected radio-on-time of node C can be easily calculated as follows.

$$E(T_{rot}^C(\text{COPE-U1})) = \frac{nT_e}{q_{AC}} + T_{csma} + T_{dar}. \tag{4}$$

The expected radio-on-time of node A depends on when node C receives sufficient encoded packets. In Figure 6, it is actually the time t_{C-ack}. The expected value of t_{C-ack} should equally divide the area of the feasible region of Case 1 (i.e., the triangle). Therefore, we can calculate the expected radio-on-time of node A as follows.

$$E(T_{rot}^A(\text{COPE-U1})) = \frac{nT_e}{q_{AC}} + \frac{2-\sqrt{2}}{2}(T_w + \frac{nT_e}{q_{AB}} - \frac{nT_e}{q_{AC}}) + T_{csma}. \tag{5}$$

Then we calculate the expected radio-on-time of node B. There are further two difference cases. First, when node B wakes up after node C has received sufficient encoded packets (i.e., $t_{B-ack} - nT_e/q_{AB} > t_{C-ack}$), its radio-on-time is just T_{cca}. In Figure 6, this first case is actually the part of shadow, which is above the line $t_{B-ack} = t_{C-ack} + nT_e/q_{AB}$. The probability of this case is its ratio in the whole triangle. Second, when node B wakes up before node C has finished packets receiving (i.e., $t_{B-ack} - nT_e/q_{AB} \leq t_{C-ack}$), its radio-on-time can be calculated as $nT_e/q_{AB} - (t_{B-ack} - t_{C-ack})$. Considering these two cases, the following equation gives the expected radio-on-time of node B.

$$\begin{aligned} E(T_{rot}^B(\text{COPE-U1})) = {} & T_{cca}\frac{(T_w - nT_e/q_{AC})^2}{(T_w + nT_e/q_{AB} - nT_e/q_{AC})^2} \\ & + ((T_{cca} + T_{dar})\frac{nT_e}{6q_{AB}} + \frac{3\frac{nT_e}{q_{AB}}T_w + 2(\frac{nT_e}{q_{AB}})^2 - 3\frac{nT_e}{q_{AB}}\frac{nT_e}{q_{AC}}}{6T_w + 3\frac{nT_e}{q_{AB}} - 6\frac{nT_e}{q_{AC}}}) \\ & * (1 - \frac{(T_w - nT_e/q_{AC})^2}{(T_w + nT_e/q_{AB} - nT_e/q_{AC})^2}). \end{aligned} \tag{6}$$

Due to the page limit, we omit the detailed derivation of the above equation. Then the overall radio-on-time of Case 1 can be calculated as $(E(T_{rot}^A(\text{COPE})) + E(T_{rot}^B(\text{COPE})) + E(T_{rot}^C(\text{COPE})))$.

Case 2. We first calculate the expected radio-on-time of node A in Case 2. Similar to Case 1, the following equation gives the expected radio-on-time of node A in Case 2.

$$E(T_{rot}^A(\text{COPE-U2})) = \frac{\sqrt{2}}{2}(T_w + \frac{nT_e}{q_{AC}} - \frac{nT_e}{q_{AB}}) + \frac{nT_e}{q_{AB}} + T_{csma}. \tag{7}$$

In order to simplify the calculation of the expected radio-on-time of node B and C, we assume that node C wakes up after node has finished receiving encoded packets from node A. When node C wakes up earlier, the radio-on-time of node B and C can be shorter. Therefore, we can calculate an upper bound of the radio-on-time of these two nodes by the above assumption. Therefore, the following equation gives the expected radio-on-time of node B and C.

$$\begin{aligned} & E(T_{rot}^B(\text{COPE-U2})) = T_{cca} + \frac{nT_e}{q_{AB}} + T_{csma} + \frac{T_w - \frac{nT_e}{q_{AB}}}{3} + \frac{nT_e}{q_{BC}}, \\ & E(T_{rot}^C(\text{COPE-U2})) = T_{cca} + \frac{nT_e}{q_{BC}} + T_{dar}. \end{aligned} \tag{8}$$

Based on the expected radio-on-time of Case 1 and Case 2 and the probabilities of these two cases, we can obtain the expected unicast radio-on-time of COPE as follows.

$$\begin{aligned} E(T_{rot}(COPE-U)) = {} & P(\text{Case-1})E(T_{rot}(\text{COPE-U1})) \\ & + (1 - P(\text{Case-1}))E(T_{rot}(\text{COPE-U2})). \end{aligned} \tag{9}$$

B. Broadcast

Consider one sender A wants to broadcasts n native packets to its m neighboring nodes R_1–R_m. We calculate the total radio-on-time of these $m+1$ nodes during the broadcasting. As shown in Figure 4 in which $m = 2$, we first consider the broadcast without using COPE. By assuming that reliable transmission is not required, we can get a lower bound of the

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/TII.2015.2412093, IEEE Transactions on Industrial Informatics

TABLE I
PARAMETER SETTING

Parameter	Default value
Wake up interval T_w	512 ms
# of native packets n	6
q_{AB} in unicast	0.8
q_{AC} in unicast	0.6
q_{BC} in unicast	0.8
# of receivers in broadcast m	5
α in broadcast m	1.5

radio-on-time without using COPE. The following equation gives the result.

$$E(T_{rot}) = T_{csma} + nT_w + mn(T_{cca} + T_e + T_{dar}). \quad (10)$$

Then we calculate the expected radio-on-time of broadcasting m packets using COPE. Compared with unicast, the radio-on-time calculation of broadcast is much simpler. The broadcasting time of the sender A is given in Equation 1, which is $(T_w + \alpha n T_e)$. Therefore, the radio-on-time of node A is $T_{csma} + (T_w + \alpha n T_e)$. The expected radio-on-time of the receivers depends on the link quality from node A to each receiver. For a receiver R_i, its expected radio-on-time can be calculated as $E(T_{rot}^{\mathrm{R}_i}(\text{COPE-B})) = nT_e/q_{\mathrm{AR}_i} + T_{cca} + T_{csma}$. Therefore, the following equation gives the total radio-on-time of broadcast.

$$E(T_{rot}(\text{COPE-B})) = T_{csma} + (T_w + \alpha n T_e) + \sum_{i=1}^{m} \frac{nT_e}{q_{\mathrm{AR}_i}} + mT_{cca} + mT_{dar}. \quad (11)$$

C. Analytical Results

We then present some analytical results based on the proposed analytical model. For unicast, we report the total radio-on-time when we tune the wake up interval T_w, the number of native packets n, q_{AB} and q_{AC}. For broadcast, we report the total radio-on-time when we tune the wake up interval T_w, the number of native packets n, the number of receivers and the constant α which affects the broadcast length. The default parameter setting is shown in Table I.

Figure 7 shows the analytical results of unicast, with COPE and without COPE. When the wake up interval T_w increases, the radio-on-time of the two methods (i.e., with COPE and without COPE) also increase. But the radio-on-time of COPE is shorter (25.84%∼32.36%) than the case without COPE and increases slower when T_w increases. When the number of native packets increases, the radio-on-time of the two methods both increases slowly and COPE reduces the radio-on-time by 25.57%∼32.21%. When the link quality of link AB increases, both of the two methods have smaller radio-on-time and COPE reduces the radio-on-time by 27.90%∼30.31%. When the link quality of link AC increases, the radio-on-time of COPE decreases. Since link AC is not used to deliver data packets in the method without COPE, its radio-on-time does not change. Note that when the link quality of link AC becomes poorer than 0.3, the performance difference of these two methods will become smaller.

Figure 8 shows the analytical results of broadcast, with COPE and without COPE. When the wake up interval T_w increases, the radio-on-time of the two methods also increase. But the radio-on-time of COPE is much shorter (81.92%∼82.98%) and increases slower than the case without COPE. When the number of native packets increases, the radio-on-time of the two methods both increases and COPE reduces the radio-on-time by 55.48%∼87.71%. When there are more receivers, the radio-on-time of the two methods both increase slowly, but COPE reduces the radio-on-time significantly (82.18%∼82.25%). The radio-on-time of the method without COPE is independent to α. The radio-on-time of COPE increase slowly when α increases and COPE reduces the radio-on-time by 82.11%∼82.54%.

VI. EVALUATION

We evaluate the energy efficiency of COPE in a testbed with 80 nodes. Since radio operations cost most of the energy, we use *radio-on-time* as a metric of energy efficiency. We use both micro-benchmarks and macro-benchmarks to evaluate COPE. In micro-benchmarks, we evaluate the unicast and broadcast efficiency of COPE by two small networks. In macro-benchmarks, we use a 80 nodes network running data collection to evaluate the energy efficiency of COPE. Collection Tree Protocol [29] is used as the routing protocol.

A. Implementation

We implement COPE in TelosB/TinyOS2.1.1 platform. A number of native data packets (e.g., n packets) are encoded by COPE at the sender side. The encoded packets are stored in a buffer. For each encoded packet, it is calculated by a coefficient vector. The receiver also needs to know the coefficient vector for decoding each encoded packet. In order to reduce the message overhead, COPE uses the same random number generator to obtain the coefficient vector, at both the sender side and the receiver side. Then each encoded packet is attached with a random seed corresponding to its coefficient vector. The receiver can use the random seed to recover the coefficient vector and perform decoding. Besides the random seed, each encoded packet also include the number of native packets been encoded, i.e., n. It is used by the receiver to decide whether it has received sufficient encoded packets. Therefore, the message overhead in each encoded packet is the random seed and the number of native packets been encoded.

B. Micro-benchmarks

Unicast. We use a small network with three nodes A, B and C (as shown in Figure 3) to evaluate the radio-on-time. The parameter setting is as follows, the wake up interval T_w is one second, the number of native packets in each round is five; 30 rounds of packet delivery are conducted. *Broadcast.* We use a sender A to broadcast packets to two receivers B, C (as shown in Figure 5) and report the radio-on-time. The parameter setting is as follows, the wake up interval T_w is one second, the number of native packets in each round is five; 10 rounds of packet delivery are conducted; α is set to be two.

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/TII.2015.2412093, IEEE Transactions on Industrial Informatics

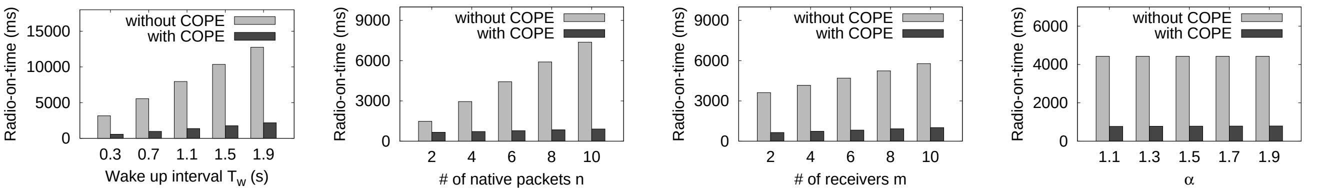


Fig. 8. Analytical results of broadcast. This figure shows the impacts of the wake up interval T_w, the number of native packets n, the number of receivers and the constant α which affects the broadcast length.

TABLE II
RADIO-ON-TIME COMPARISON IN THE MICRO-BENCHMARKS

	Method/Node	A	B	C	Total
Unicast	Without COPE (ms)	12132	11109	2405	25646
	With COPE (ms)	10420	8034	2395	20849
	Radio-on-time reduction	14.1%	27.7%	0.4%	18.7%
Broadcast	Method/Node	A	B	C	Total
	Without COPE (ms)	59388	2232	2230	63850
	With COPE (ms)	12469	1693	1689	15851
	Radio-on-time reduction	79.0%	24.1%	24.3%	75.2%

Table II shows the results of the micro-benchmarks. In unicast, COPE reduces 18.7% of the total radio-on-time. In particular, radio-on-time of node B is reduced at most. The reason is as follows. In COPE, node C has the chance to receive sufficient encoded packets before node. In this case (i.e., Case 1 of unicast), node B can return to sleep immediately and its radio-on-time is reduced significantly. In broadcast, COPE reduces the total radio-on-time by 75.2%. The energy saving is mainly from the sender A. The reason is that in COPE, node A can broadcast multiple encoded packet in one wake up interval. These results show that COPE is able to improve the energy efficiency of both unicast and broadcast.

C. Macro-benchmarks

80 TelosB nodes (shown in Figure 9) running data collection task are used to evaluate the energy saving performance of COPE. These nodes form an 8*10 grid and the internode distance is 60 cm. The power level of each node is tuned down to form a multihop network. The sink locates at the corner. Each node generates packets at the similar time (with a one second randomness) in the experiments. The packets are delivered to the sink by a typical collection protocol [29]. Three approaches are evaluated in these experiments, COPE, BoXMAC, and a typical low power opportunistic routing approaches ORW. We run the experiments for five times and report the average values. Figure 10 shows the radio-on-time of these 80 nodes in the three different approaches. On average, the radio-on-time of COPE is 32407 ms, the radio-on-time of BoXMAC is 46412 ms, the radio-on-time of ORW is 38799 ms. Compared with BoXMAC, COPE delivers data by preamble data more efficiently. Compared with ORW, COPE reduces the number of acknowledges significantly. Therefore, the COPE achieves the higher energy efficiency.

Fig. 9. The test-bed with 80 TelosB nodes.

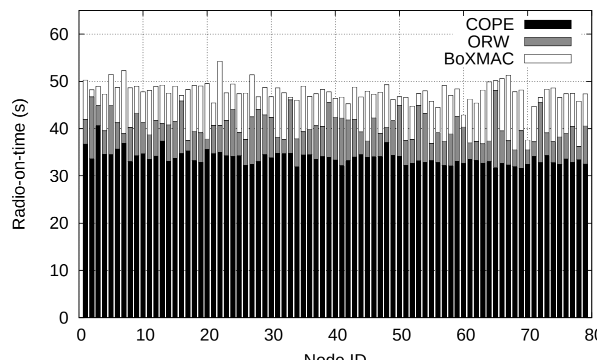


Fig. 10. The radio-on-time of 80 nodes running COPE, ORW, and BoXMAC.

VII. CONCLUSION

In this paper, we propose COPE, a coded preamble design for low power wireless sensor networks. COPE exploits the data delivery potential of preambles in asynchronized low power MAC protocols, by using network coding to encode the preambles packets. COPE is a versatile 2.5 layer between the low power link layer (layer 2) and the network layer (layer 3), supporting multiple upper layer routing primitives, such as unicast and broadcast. Results of theoretical analysis and experiments show that COPE is able to significantly improve the energy efficiency under various network settings.

REFERENCES

[1] L. Mo, Y. He, Y. Liu, J. Zhao, S. Tang, X. Li, and G. Dai, "Canopy Closure Estimates with GreenOrbs: Sustainable Sensing in the Forest," in *Proceedings of ACM SenSys*, 2009.

[2] M. Magno, D. Boyle, D. Brunelli, E. Popovici, and L. Benini, "Ensuring survivability of resource-intensive sensor networks through ultra-low power overlays," *Industrial Informatics, IEEE Transactions on*, vol. 10, no. 2, pp. 946–956, 2014.

[3] J. Polastre, R. Szewczyk, and D. Culler, "Telos: Enabling Ultra-Low Power Wireless Research," in *Proceedings of IPSN*, 2005.

[4] G. Anastasi, M. Conti, and M. Di Francesco, "Extending the lifetime of wireless sensor networks through adaptive sleep," *Industrial Informatics, IEEE Transactions on*, vol. 5, no. 3, pp. 351–365, 2009.

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/TII.2015.2412093, IEEE Transactions on Industrial Informatics

[5] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient mac protocol for wireless sensor networks," in *Proceedings of IEEE INFOCOM*, 2002.

[6] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proceedings of ACM SenSys*, 2004.

[7] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks," in *Proceedings of ACM SenSys*, 2006.

[8] A. Castagnetti, A. Pegatoquet, T. N. Le, and M. Auguin, "A joint duty-cycle and transmission power management for energy harvesting wsn," *Industrial Informatics, IEEE Transactions on*, vol. 10, no. 2, pp. 928–936, 2014.

[9] D. Moss and P. Levis, "BoX-MACs: Exploiting Physical and Link Layer Boundaries in Low-Power Networking," Stanford, Tech. Rep., 2008.

[10] G. Anastasi, M. Conti, and M. Di Francesco, "A comprehensive analysis of the mac unreliability problem in ieee 802.15.4 wireless sensor networks," *Industrial Informatics, IEEE Transactions on*, vol. 7, no. 1, pp. 52–65, 2011.

[11] P. Dutta, S. Dawson-Haggerty, Y. Chen, C.-J. M. Liang, and A. Terzis, "Design and evaluation of a versatile and efficient receiver-initiated link layer for low-power wireless," in *Proceedings of ACM SenSys*, 2010.

[12] J. Wang, W. Dong, Z. Cao, and Y. Liu, "On the delay performance in a large-scale wireless sensor network: Measurement, analysis, and implications," *Networking, IEEE/ACM Transactions on*, vol. 23, no. 1, pp. 186–197, 2015.

[13] P. T. A. Quang and D.-S. Kim, "Throughput-aware routing for industrial sensor networks: Application to isa100.11a," *Industrial Informatics, IEEE Transactions on*, vol. 10, no. 1, pp. 351–363, 2014.

[14] J. Niu, L. Cheng, Y. Gu, L. Shu, and S. Das, "R3e: Reliable reactive routing enhancement for wireless sensor networks," *Industrial Informatics, IEEE Transactions on*, vol. 10, no. 1, pp. 784–794, 2014.

[15] C. Zhu, L. Yang, L. Shu, V. Leung, T. Hara, and S. Nishio, "Insights of top- k query in duty-cycled wireless sensor networks," *Industrial Electronics, IEEE Transactions on*, vol. 62, no. 2, pp. 1317–1328, 2015.

[16] H. Le, J. van Eck, and M. Takizawa, "An efficient hybrid medium access control technique for digital ecosystems," *Industrial Electronics, IEEE Transactions on*, vol. 60, no. 3, pp. 1070–1076, 2013.

[17] T. Kim, I. Kim, Y. Sun, and Z. Jin, "Physical layer and medium access control design in energy efficient sensor networks: An overview," *Industrial Informatics, IEEE Transactions on*, vol. 11, no. 1, pp. 2–15, 2015.

[18] Y. Sun, O. Gurewitz, and D. B. Johnson, "Ri-mac: a receiver-initiated asynchronous duty cycle mac protocol for dynamic traffic loads in wireless sensor networks," in *Proceedings of ACM SenSys*, 2008.

[19] C. Zhu, L. Yang, L. Shu, V. Leung, J. Rodrigues, and L. Wang, "Sleep scheduling for geographic routing in duty-cycled mobile sensor networks," *Industrial Electronics, IEEE Transactions on*, vol. 61, no. 11, pp. 6346–6355, 2014.

[20] O. Landsiedel, E. Ghadimi, S. Duquennoy, and M. Johansson, "Low power, low delay: Opportunistic routing meets duty cycling," in *Proceedings of the IPSN*, 2012.

[21] D. Zhang, G. Li, K. Zheng, X. Ming, and Z.-H. Pan, "An energy-balanced routing method based on forward-aware factor for wireless sensor networks," *Industrial Informatics, IEEE Transactions on*, vol. 10, no. 1, pp. 766–773, 2014.

[22] J. Heo, J. Hong, and Y. Cho, "Earq: Energy aware routing for real-time and reliable communication in wireless industrial sensor networks," *Industrial Informatics, IEEE Transactions on*, vol. 5, no. 1, pp. 3–11, 2009.

[23] J. Luo, J. Hu, D. Wu, and R. Li, "Opportunistic routing algorithm for relay node selection in wireless sensor networks," *Industrial Informatics, IEEE Transactions on*, vol. 11, no. 1, pp. 112–121, 2015.

[24] A. Hagedorn, D. Starobinski, and A. Trachtenberg, "Rateless Deluge: Over-the-Air Programming of Wireless Sensor Networks using Random Linear Codes," in *Proceedings of IPSN*, 2008.

[25] Y. Gao, J. Bu, W. Dong, C. Chen, L. Rao, and X. Liu, "Exploiting concurrency for efficient dissemination in wireless sensor networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 24, no. 4, pp. 691–700, Apr. 2013.

[26] W. Dong, C. Chen, X. Liu, J. Bu, and Y. Gao, "A lightweight and density-aware reprogramming protocol for wireless sensor networks," *Mobile Computing, IEEE Transactions on*, vol. 10, no. 10, pp. 1403–1415, Oct. 2011.

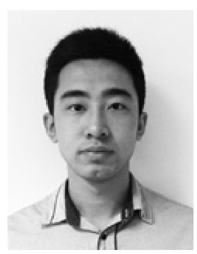
[27] R. Chandanala, W. Zhang, R. Stoleru, and M. Won, "On combining network coding with duty-cycling in flood-based wireless sensor networks," *Ad Hoc Networks*, vol. 11, no. 1, pp. 490–507, 2013.

[28] "Fountain codes," *Communications, IEE Proceedings*, vol. 152, no. 6, pp. 1062–1068, 2005.

[29] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection tree protocol," in *Proceedings of ACM SenSys*, 2009.

Yi Gao (S'09-M'15) received the B.S. and Ph.D. degrees in Zhejiang University in 2009 and 2014, respectively. He is currently a research assistant professor in Zhejiang University, China. From December 2008 to April 2009, he worked in the Information System College of Singapore Management University as an exchange student. From October 2011 to October 2012, he worded in McGill University as a joint training research student. His research interests include protocols design and measurement in networks. He is a member of the IEEE.

Wei Dong (S'08-M'11) received the B.S. and Ph.D. degrees in Computer Science from Zhejiang University in 2005 and 2010, respectively. He was a Postdoc Fellow at the Department of Computer Science and Engineering in Hong Kong University of Science and Technology from Dec. 2010 to Dec. 2011. He is currently an associate professor at the College of Computer Science in Zhejiang University. His research interests include networked embedded systems and wireless sensor networks.



Lizhi Deng received his B.S. degree in 2013. He is currently a graduate student in the College of Computer Science of Zhejiang University. His research interests include protocol design and implementation in wireless sensor networks.

Chun Chen (M'13) received his Bachelor of Mathematics degree from Xiamen University, China, in 1981, and his M.S. and Ph.D. degrees in Computer Science from Zhejiang University, China, in 1984 and 1990 respectively. He is a professor in College of Computer Science, and the Director of Institute of Computer Software at Zhejiang University. His research interests include embedded system, image processing, computer vision, and CAD/CAM.

Jiajun Bu (M'06) received the B.S. and Ph.D. degrees in Computer Science from Zhejiang University, China, in 1995 and 2000, respectively. He is a professor in College of Computer Science and the deputy director of the Institute of Computer Software at Zhejiang University. His research interests include embedded system, mobile multimedia, and data mining. He is a member of the IEEE and the ACM.