A Joint MAC and Routing Approach for Duty-cycled Wireless Sensor Networks

Masato Yokotani
Graduate School of Systems Engineering,
Wakayama University,
930 Sakaedani, Wakayama, 640-8510, Japan
e-mail: yokotani.masato@g.wakayama-u.jp

Takuya Yoshihiro
Faculty of Systems Engineering,
Wakayama University,
930 Sakaedani, Wakayama, 640-8510, Japan
e-mail: tac@sys.wakayama-u.ac.jp

***Abstract*—We propose a new joint MAC and Routing Approach for duty-cycled sensor networks and designed a protocol called JBS (Joint MAC and routing protocol for Beacon-Saving), which achieves far larger network lifetime than conventional ones. Our proposal improves energy performance of receiver-initiated MAC protocols by combining it with routing functionality. In receiver-initiated MAC, leaf nodes of the delivery tree periodically transmit wasteful beacons even though no data frame will be received. So, we propose to omit transmitting beacons of leaf nodes to save the transmission power as well as the accompanying idle listening. As revealed in the past studies, reduction of idle listening has large effect on the energy performance. In our protocol, we build a delivery tree in a distributed manner such that the number of relay nodes are minimized, meaning that we maximize the number of leaf node. When residual battery power of some relay node becomes low, the delivery tree is recomposed so that every relay node in it is with sufficient residual power. Evaluation results show that the effect of our approach to combine routing functionality is far larger than the power saving effect of MAC only.**

I. Introduction

Wide range of applications are waiting for the technology for battery-powered wireless sensor networks that suffice the practical requirements. Low power consumption to achieve long enough lifetime is one of the most focused property as the requirements for sensor networks. Several protocols such as IEEE802.15.4 and Bluetooth have been standardized for multi-hop sensor networks. However, they generally work at most several months if they work as multi-hop networks, which do not suffice the practical requirement in many cases.

To enhance the lifetime of sensor networks for delay tolerant applications such as environmental sensing etc., many MAC protocols including B-MAC [11], X-MAC [12], RI-MAC [10], A-MAC [14], etc., have been developed in the literature. Those duty-cycle MAC protocols significantly improve the lifetime of sensor networks, but they do not still suffice the lifetime that meets the requirements of practical deployment. Ideally, once a sensor node is placed in a field, it is expected to work at least 5-10 years since the maintenance cost for massive sensors is the great matter in deploying sensor networks. The series of duty-cycle MAC studies revealed that the bottleneck is the idle listening; a certain amount of idle listening time in duty-cycle MAC is necessary to surely receive frames under presence of back-off contention and time drift on clocks.

Wake-up Radio (WuR) has appeared and focused on as a technique to overcome this bottleneck, and comparative studies showed that WuR significantly prolong the lifetime of sensor nodes [4]. WuR system enables sensor nodes to wake up without listening radio, or with listening in very low power consumption, by using special hardware circuits to identify the wake-up signals. Because WuR directly wake up specific nodes without idle listening, it dramatically reduces power consumption and enhances the lifetime of sensor nodes.

Unfortunately, WuR has a drawback that they require strong radio signal to wake up nodes. This means that the distance between nodes must be considerably small, or transmission power must be very strong, which poses a severe limitation in the practical deployment. Additionally, we note that the transmission power in ISM bands such as 2.4GHz, 5GHz, and Sub-1GHz bands are usually limited by the law of each countries because strong signals will bring severe interference in wide areas of lands. Consequently, as both duty-cycle MAC and WuR have their own drawbacks, we would have choice on which type of sensor systems to use according to the requirements of applications. Roughly speaking, duty-cycle MAC is for delay-tolerant applications with relatively sparse deployment, whereas WuR is for delay-sensitive applications with dense deployment.

In this paper, we again focus on duty-cycle MAC and propose a new approach to reduce power consumption of multi-hop sensor networks. We propose to combine the receiver-initiated MAC protocols such as RI-MAC with routing functionarily. The main idea comes from the fact that the leaf nodes of the data delivery tree do not need to receive data frames. So, in our proposal, we construct the data delivery tree such that the number of leaf nodes is maximized, and omit beacons as well as idle listening of the leaf nodes. Additionally, we introduced the mechanism to adaptively reconstruct the delivery tree; when some relay nodes of the tree mostly run out of battery, we adaptively modify the tree such that all relay nodes in the new delivery tree have sufficient battery power. To enable nodes to exchange control messages in reconstructing the tree even if leaf nodes omits beacons, we keep a part of the leaf nodes being ready to receive frames. In this way, our new approach achieves both low power consumption and adaptive topology managements at the same time to be feasible in practical deployment.

The reminder of this paper is as follows. In Sec.II, we describe the related work on saving power consumption of multi-hop sensor networks. In Sec.III, we present the proposed joint MAC and routing protocols. In Sec.IV, we give the evaluation results on our protocols. Finally, we conclude the work in Sec.V.

II. Related Work

A. Duty-cycle MACs

Duty-cycle MACs control the wake-up and sleep timing of sensor nodes so as to synchronize the working time of sender and receiver nodes while reducing the energy consumption. A large volume of duty-cycle MAC protocols has been accumulated in the literature so that they are sometimes called "MAC alphabet of soups" [6]. The main challenge in designing duty-cycle MAC lies in how to synchronize senders and receivers with as low power consumption as possible. As representative proposals in the early stage of the study, B-MAC, proposed by Polastre et al., transmits preamble longer than the duty-cycle interval to make receivers surely perceive the transmission of senders [11]. X-MAC by Buettner et al. transmits repeated short preambles instead of long continuous one to improve energy efficiency [12]. However, those sender-initiated MACs suffer from large energy consumption in transmitting and receiving preambles.

Later, an improved approach of receiver-initiated MAC appears in which receivers transmit periodical beacons to notify senders that they are ready to receive frames. Sum et al., proposed RI-MAC that start back-off competition of senders after receiver's beacons [10]. Huang et al., proposed RC-MAC that improved it by enabling receivers to designate the next sender in beacons according to the traffic patterns they learned [13]. Dutta et al. proposed A-MAC that introduced a technique called Auto-Ack to further save energy by letting receivers know existence of contending senders before sending data frames [14]. A large volume of followers exists for further improvement of duty-cycle MAC protocols shown in several survey articles such as [2]. However, duty-cycle MACs essentially require idle-listening to absorb the time drift of wake-up timing, which gives the limitation of duty-cycle MAC protocols [4].

B. Wake-up Radio (WuR)

WuR is a hardware-assisted mechanism to eliminate or significantly reduce the power consumption of idle-listening [3]. Receiver nodes equip the extra WuR antenna and circuit to identify the wake-up signal coming from sender nodes. WuR harvests power from the transmitted radio to process the wake-up signal and trigger for waking-up the main chip when it detects the wake-up signal targeting on the node. Since WuR detects wake-up signals with very low current, it significantly reduces the wasteful power consumption of idle listening.

However, the weakness of WuR exists in the distance that the wake-up signals reach. Because the efficiency of energy harvesting from radio is low, the WuR receivers require large signal power to process it. Although active-type WuR assisted by battery has higher sensitivity, reachable distance under the same transmission power is far shorter than normal data communications. For instance, according to the survey article [3] written by Djiroun, et al., current WuR prototypes wake-up nodes within 10-40[m] in case of 10[dBm] transmission power under sub-1GHz band, whereas normal data communications are possible within 100-200[m] in general. Also, transmission power is often legally limited by a threshold value determined independently by countries.

C. Energy Saving from Routing Approach

To extend the lifetime of sensor networks, various network-layer techniques have been proposed that controls forwarding paths of data packets to reduce power consumption so as to extend the lifetime of networks. Luo et al., proposed a centralized algorithm to compute the optimal delivery tree that maximize the lifetime of networks [16]. Kuo et al., minimizes the total power consumption of sensor networks for indoor powered sensor networks [17]. Rodoplu et al., computes the optimal transmission power of each node to form delivery tree that minimizes power consumption [5]. Also, several routing protocols for load balancing that intend to consume power of all nodes equally under dynamic topology changes have been proposed [8] [9]. However, their effect is quite limited because they only depend on routing issues whereas MAC or duty cycling schedules actually dominate the power consumption.

Only a few studies exist that treat joint MAC and routing protocols. For instance, JAM [1] proposed by Li et al. dynamically adjusts MAC and forwarding parameters within fixed data delivery tree to prolong the lifetime of sensor networks. To the best of our knowledge, collaboration between MAC behavior and delivery-tree computation is the first approach appeared in the literature.

III. JBS: The Proposed Scheme

A. Overview

We propose to modify receiver-initiated MAC protocols for sensor networks such as RI-MAC to save energy consumption by omitting beacon transmission on leaf nodes of the data delivery tree. To this end, collaboration between MAC and routing protocols are required to change MAC behaviors under dynamic topology changes due to failure or battery shortage of nodes. We assume a sensor network in which every node periodically measures values from its sensors to be collected to sink nodes. Also, we assume the transmission power of nodes is constant, and the density of nodes is high enough to afford multiple paths from each node to a sink node. Our protocol supports multiple sinks, although our protocol description and examples treat single-sink cases for simplicity.

First, in JBS, we introduce two states called *stable* state and *construction* state; nodes transit to *construction* state when node failure or topology changes occur to reconstruct data delivery tree, and transit to *stable* state if reconstruction of the delivery tree finishes. In *construction* state, every node (including leaf nodes) periodically transmits beacons that includes messages of routing protocols so that the routing protocol works to construct or modify the data delivery tree. On the other side, in *stable* state, leaf nodes omit MAC beacons to save power whereas the other nodes, i.e., relay nodes, periodically transmit beacons since they should relay data traffic. Nodes save power consumption in *stable* state such that leaf nodes omit transmission of beacons, and transit to *construction* state to adjust the delivery tree only when required due to node failure or topology changes.

Second, we designed the routing protocol to build the delivery tree such that the number of relay nodes is minimized in a distributed manner to save power consumption as much as possible. Note that this is done by making data traffic concentrated to specific nodes, which does not appear in the conventional power-saving routing approaches; Most of them intends to consume power of nodes as equally as possible over the network. When node failure or topology changes occur, it is conveyed to nodes along the delivery tree in the hop-by-hop manner in both direction to the parent and to the children. Here, JBS tries local repair of the delivery tree to save power consumption so that the minimum required number of nodes gets to know the topology change and transits to *construction* state. Although the repaired tree may not be the optimal one in terms of the number of relay nodes, local repair policy saves the power consumption in practice.

Third, note that local repair procedure sometimes does not work with only parent-children communications along the delivery tree; if a new path want to go through another branch of the delivery tree, a node need to contact neighbors that are not neither its parent or a child of it. However, the neighbor may be a leaf node that is not ready to receive packets because it does not transmit any beacons. To cope with this situation, JBS prepares the nodes called *reserved relay* that do not omit beacons and behaves the same as relay nodes even if it is a leaf node. By preparing the *reserved relay* nodes dense enough to provide detour routes in the local repair procedure, JBS is enabled to adaptively change paths under dynamic environments.

Finally, to save power consumption and prolong the lifetime of the network, JBS triggers local repair if the residual battery power becomes lower than the preconfigured threshold. By adaptively triggering the local repair procedure according to the residual battery power, we maintain the delivery tree such that relay nodes are always with sufficient battery power, which prolongs the lifetime of sensor networks.

B. Introducing States of Nodes

JBS works based on the receiver-initiated MAC protocol such as RI-MAC. Although we can use any receiver-initiated MAC, we in this paper assume RI-MAC for conciseness. In RI-MAC, nodes periodically transmit beacons that notify neighbors of being ready to receive frames. After receiving a beacon, sender nodes invoke random back-off then transmit frames to the receiver. If contending senders exist such as n_{s1} and n_{s2} in the figure, this back-off-based transmission process continues, and when no signal is detected within the back-off window, the receiver quits waiting and goes to sleep. Because senders are possible to learn the timing of receiver's beacons, power consumption for idle listening is reduced significantly.

Note that every node transmits beacons to receive frames from other nodes. However, in case of duty-cycled sensor networks, leaf nodes in the data delivery tree have no need to receive data frames. Nodes actually need to receive control frames of routing protocols in case of topology changes, but the frequency of topology changes is not high in reality. Thus, JBS distinguishes the state of nodes, i.e., *stable* and *construction* states, and switch them in order to save power consumption when topology changes are not occurring.

The *stable* state is the static low-energy state in which the delivery tree is stably unchanged. In the *stable* state, a *leaf* node, which is located at a leaf position of the delivery tree, just has to send the periodically sensed values to its next-hop, and does not need to receive neither data nor control frames. So, leaf nodes in *stable* state omit beacons to save power consumption. As a result, in return for that the receiver nodes are prohibited to receive any frames except Ack frames transmitted from next-hop node, their energy consumption and lifetime is dramatically improved by omitting beacons as well as the following idle listening, which occupies a considerable part of energy consumption in *leaf* nodes. Note that, since *relay* nodes in the delivery tree need to receive data frames to relay packets to sink nodes, they cannot stop periodical beacons, so they behave the same as RI-MAC.

In contrast, *construction* state is the dynamic state in which the routing protocol works to build a new delivery tree by changing next-hops of each node. In this state, all nodes behave the same as RI-MAC except that nodes are always awake and do not sleep. Consequently, every node in *construction* state are ready to transmit/receive frames so that nodes are able to broadcast routing messages to neighbors. Because *construction* state consumes large energy due to long awaking time, nodes transit to *stable* state when *construction* state lasts for preconfigured time and the delivery tree is expected to be converged.

C. Routing Protocol to Reduce Relay Nodes

JBS largely prolongs the lifetime of leaf nodes by omitting beacon transmissions. In contrast, relay nodes have still short lifetime because they cannot omit beacon transmissions. To lead to prolong lifetime of networks, we first reduce the total power consumption of all nodes by building a delivery tree with smaller number of relay nodes.

The routing protocol works when nodes are in *construction* state. In this state, each node includes routing messages in beacons, which is transmitted periodically to inform neighbor nodes. First of all, each node n gets to know its distance $level(n)$, which is the distance in hop-count from a nearest sink node. Sink nodes, which also behave the same as sensor nodes, advertise periodically the distance 0 in their routing messages. Nodes that do not know their distance advertise the distance ∞ similarly. Then, a node n determines its distance as $level(n) = m + 1$ where m is the smallest value among

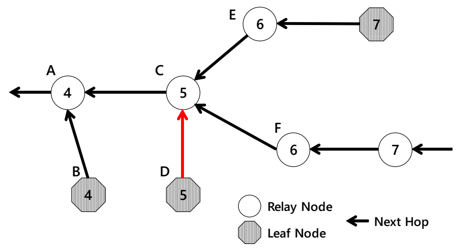


Fig. 1. Example of Next-hop Selection

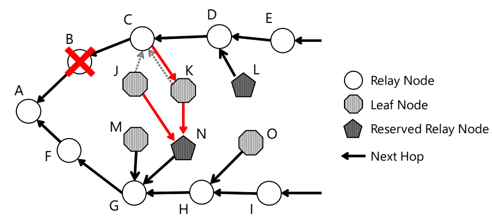


Fig. 2. Local Reconstruction with Reserved Relays

all distances in the routing messages received from neighbors. Specifically, when a node n receives a new routing message from a neighbor, it updates its distance $level(n)$ if the distance reported by the message is less than $level(n) - 1$. After the sufficient number of repetition of this process, every node finally obtains the correct distance from a sink node as long as it has connection to at least one sink node.

Next, each node selects its next-hop node to forward data packets. In addition to $level(n)$, n also periodically broadcast in the routing messages its next-hop node ID $par(n)$ and the number of descendant $desc(n)$ in the delivery tree. The latter can be counted as the sum of $desc(\cdot)$ values informed from the nodes that select n as their next-hop. After receiving routing messages from all neighbors, n selects as its next-hop the node with the maximum number of descendants. Here, to avoid routing loops, n must select its next-hop from the set of neighbors whose distance is $level(n) - 1$ if n is a *relay* node, and from those whose distance is $level(n)$ or $level(n) - 1$ if n is a *leaf* node. Note that node type is determined by whether there is a neighbor that selects n as its next-hop. After sufficient number of messages exchange, next-hops of all node is surely converged, and the delivery tree is determined. So, each node transits to *stable* state when preconfigured time T_s passes after the last update of its next-hop.

Fig. 1 depicts the example of next-hop selection of JBS where the number on each node n represents $level(n)$ and the arrows represent the next-hop nodes. Every node n that has a child whose next-hop is n (e.g., A, C, E, F) is a *relay* node, which selects next-hop from neighbors whose distance is $level(n) - 1$. In contrast, *leaf* node such as D can select the same-level node C as its next-hop. If D selects C instead of B as its next-hop, we can reduce the number of *relay* nodes by keeping B as a *leaf* node.

D. Local Reconstruction of Data Delivery Tree

To reconstruct the delivery tree adaptively according to the topology-change events such as node failure, nodes must transit to *construction* state. If we intend to build the optimal delivery tree, i.e., if we expect the same delivery tree build from scratch, in face of topology-change event, it is natural to expect all nodes to transit to *construction* state. However, this is inefficient in terms of power consumption since *construction* state consumes large power due to long awaking time. So, we designed JBS to adopt local reconstruction of the delivery tree so that only the small part of nodes are going to be in *construction* state in face of local topology-change events such as node failure.

Notice that, in *stable* state, topology-change events can be propagated among nodes only along the delivery tree via Data or Ack frames since *leaf* nodes do not transmit beacons so that *leaf* nodes except for direct children cannot be approached via routing messages. Thus, in JBS, we prepare the *reserved relay* nodes (we call them *r-relay* hereafter) to help communications among nodes in different branches. *R-relay* nodes behave the same as *relay* nodes (so they transmit periodical beacons) even though no node selects them as their next-hops. By locating *r-relay* nodes between branches with an appropreate interval, we can use extra paths to notify events to other nodes. Namely, *r-relay* enables nodes to share topology-change events without necessarily passing messages along the delivery tree.

See Fig. 2 for example. Initially, leaf nodes J, K, and L belong to the upper branch rooted at B while M, N, and O belong to the lower branch rooted at F. Among them, L and N play the role of *r-relay* nodes. If B fails, nodes C, D, E, J, K, and L lose their path to the sink. If we do not prepare *r-relay* nodes, they cannot set up new paths to the sink because leaf nodes M, N, and O, which are neighbors of J, K, and L, cannot receive messages from others due to beacon omission. In contrast, if N and L are working as *r-relay* nodes, J and K are possible to send some messages to N in response to its beacons, and arrange to use N as their Next-hops. As a result, all nodes in the upper branch obtain the paths to the sink through N. In this way, *r-relay* nodes enable us to locally repair the delivery tree against topology changes.

E. Selection of Reserved Relays

A set of *r-relay* nodes should be selected in the appropreate interval in the network to help local communications among nodes, while the number of *r-relay* nodes should be as small as possible to save power consumption. We designed the procedure to select a set of *r-relays* such that every leaf node has at least one detour path via *r-relay* nodes that does not use the grandparent node in the delivery tree. The procedure works in fully distributed manner so that the process is kept simple and easy to implement into sensor nodes.

The procedure for *r-relay* selection also works during *construction* state. First, by collecting next-hop IDs included in routing messages, each node n creates the set of neighbors' next-hops $N_{nh}(n)$. Next, n computes $Cover(n)$, which we call the covering index (CI) of n, which is defined as the number of neighbors $m \in N(n)$ such that $m \in N_{nh}(n)$ and $m \notin D(par(par(n)))$, where $D(par(par(n)))$ is the set of descendants of n's grandparent. CI represents the number of neighbors that belongs to different branches, meaning that, the larger the CI value, the more capable to help local communications.

CI computed at each node is advertised via routing messages. Each node n also advertise the node ID $maxCov(n)$ that advertises the largest CI among the node and its neighbors. From the above information advertised, each node independently determines whether it should be *r-relay* nodes or not. The basic strategy is that, if n is selected by majority of n's neighbors m as their $maxCov(m)$ node, n becomes *r-relay*. Specifically, let M be the set of node m such that $m \in N(n)$, $maxCov(m) \in N(n)$, and $m \in D(par(par(n)))$. Also let n_{max} be the node that has the maximum CI in M. Then, if $Cover(n) > Cover(n_{max})$, n becomes *r-relay*.

As above, by exchanging $maxCov(\cdot)$ node among neighbors, nodes autonomously becomes *r-relay* nodes such that *r-relays* are surely located in a certain geometrical interval while the number of *r-relays* is as small as possible.

F. Triggering Local Reconstruction

In JBS, nodes trigger reconstruction of delivery tree when they detect abnormality in the tree such as node or link failure. If a node detects failure in its parent or children in the delivery tree, the node transits to *construction* state and start the local repair process described in the next section. Note that failure of children is detectable by observing the periodical reception of data frames from children. In addition, when a node detects that its residual battery power is lower than the preconfigured threshold C_{change}, it also triggers local repair process by notifying the low-battery event to its parent and all children, and then becomes *leaf* node. The notification is done by using flags in Data and Ack frames periodically exchanged along the delivery tree. Nodes that is notified of the low-battery event transit to *construction* state in the same way as the case of node/link failure.

Note that, if a relay node with low residual battery becomes *leaf* node, the lifetime of the node is extremely prolonged, which also leads to prolong the lifetime of the sensor networks.

When a node detects failure or the low-battery state of the neighbors in the delivery tree as described in the previous section, it transits to *construction* state and starts the local repair process. JBS repairs the tree using r-relay nodes with as small number of nodes being in *construction* state as possible. To this end, JBS controls the area of *construction*-state nodes to expand gradually as time passes.

In case of parent node failure, node n that detects failure (or low-power notification) of its parent node transits to *construction* state and notifies the children *leaf* nodes of the topology-change via the corresponding flag in beacons to make them transit to *construction* state. Note that n and its children *leaf* nodes have null next-hops and null distance at this moment in *construction* state. So, if one of them finds a valid next-hop via its neighbor *relay* or *r-relay* nodes, they can set a valid next-hop with the correct distance according to the normal behavior of *construction* state nodes. When they transits to *stable* state after waiting for T_s time as a normal process of *construction*-state nodes, reconstruction of delivery tree finishes. Otherwise, i.e., if n cannot find any valid next-hops during a preconfigured time period T_p, then n makes its children *relay* nodes transit to *construction* state. They also make its children *leaf* nodes *construction* state in the same way and wait for T_p. This process continues until the delivery tree is repaired by finding a valid next-hop.

On the other hand, in case of child node failure, node n that detects failure (or low-power notification) of one of its child just make all of their children *leaf* nodes transit to *construction* state, and return to *stable* state after waiting for time T_s without making other relay nodes *construction* state. Note that this operation enables us to use n's leaf nodes as a substitute for the failed *relay* node. Because n's leaf nodes are expected to be geometrically close to the failed node, they are suitable for replacement and should join the local repair process to repair the delivery tree with the minimum modification.

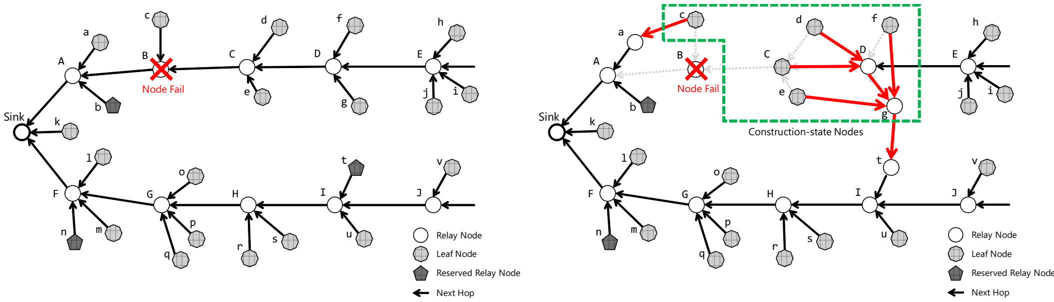
Note that nodes should not select low-power nodes as their next-hop (i.e., as a *relay* node), since it causes frequent reconstruction that leads shortened network lifetime. Thus, we introduce a threshold value C_{slimit} such that $C_{slimit} > C_{change}$, and nodes avoid to select nodes r with $Batt(r) < C_{slimit}$ where $Batt(r)$ denotes the residual battery power of r. Note that this rule is applied to the general process of JBS so that nodes with lower residual power than C_{slimit} is never selected as *relay* nodes.

G. An Example of Local Reconstruction

We present an example of local reconstruction of delivery tree. Fig. 3(a) is the initial state in which we assume that node B fails. *Relay* node C and *leaf* node c, which are children of B, detect failure of B, and transits to *construction* state with its next-hop updated with null. Simultaneously, node A also detects the failure of its child B and transits to *construction* state. Since A and C are *relay* nodes, they activate the flag in beacons and make their children *leaf* nodes a, b, d and e transit to *construction* state. As a periodically transmits beacons, c receives a's beacons and updates its next-hop with a.

On the other hand, nodes C and its children *leaf* nodes d and e cannot find any valid next-hop in this situation. Thus, after waiting for T_s time, C activates the flag on its beacons that make the child *relay* node D transit to *construction* state. After D changed its state, D makes its children *leaf* node f and g transit to *construction* state by activating flags on beacons. After D, f, and g change their state and simultaneously update their next-hops with null, g finds a *r-relay* node who has a valid next-hop t. Nodes $C, D, d, e, f,$ and g behave according to the procedure of *construction* state, and reconstruct the delivery tree as shown in Fig. 3(b). Once the new next-hop is determined, every node transits to *stable* state after waiting for time T_s, and the tree reconstruction finishes.

In this way, the delivery tree is reconstructed without using failure or low-power nodes as relay nodes. To limit



(a) Initial State

(b) After Local Reconstruction

Fig. 3. An Example of Local Reconstruction

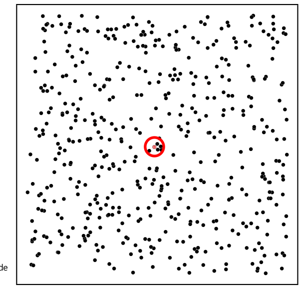


Fig. 4. Simulation Scenario ($n = 500$)

the reconstruction effect within local minimum areas, JBS gradually expands the searching area from node C to D, and E if unless no valid next-hop is available. In finding valid next-hops, *r-relay* nodes such as t play an important role. Furthermore, by changing the state of B's (i.e., failure node's) parent A and its children *leaf* nodes, JBS saves B's children such as c. This also works effective if C or d is in the communication range of c; the local reconstruction finishes with minimum changes just by replacing B with a and c.

IV. Evaluation

A. Evaluation Methods

JBS prolongs the lifetime of sensor networks by omitting beacon transmissions in receiver-initiated MAC combined with routing protocols that adaptively reconstruct delivery tree considering the residual energy of nodes. Since there is no proposal in the literature that combines receiver-initiated MAC with adaptive routing, we compare the performance of JBS with RI-MAC, where RI-MAC is combined with a simple load-balanced routing scheme. In the load-balanced routing scheme, sender nodes simply select their next-hops randomly from neighbors that has the distance just 1-hop smaller than the senders. Note that both RI-MAC and the load-balanced routing are considered to have excellent effect on prolonging network lifetime. So, by measuring the difference in energy consumption performance, we can clarify the significant effect of JBS coming from the combination of RI-MAC and the routing scheme.

Because JBS is supposed to work effectively in large sensor networks that covers a large area with up to one thousand of nodes, we selected simulation evaluation rather than real hardware implementation. To simulate the power consumption of JBS and the compared protocol, we implemented the protocols from scratch with C++ language as a event-driven simulator that exactly simulates MAC and routing behaviors including random back-off contentions in MAC and message passing in routing protocols. Note that, we judged that implementing on existing simulators is not realistic mainly in terms of simulation speed because we simulate with more than several hundreds of nodes. By omiting physical-layer behavior and the overhead for general network functions, we realized the exact simulation of the protocols from the viewpoint of energy consumption with feasible computational time. As the physical-layer model, we adopted so called Single Disk Model [15] in which communications always suceed if two nodes are within a certain physical distance. This model works in our case because we suppose sparse traffic that does not collides frequently.

For accurate estimation of power consumption, we assume a reference hardware model that reflects on up-to-date micro-controllers and RF chips for 2.4GHz sensor devices, and used the real specification parameter values from their datasheet. We describe the hardware and power consumption model in Sec. IV-C and IV-D.

B. Evaluation Scenarios

As the simulation scenario, we suppose the environmental sensing application that covers a large area to sense values such as temparatures, etc. with a certain time interval. We suppose 500×500[m] square field with n randomly-located nodes where we tried $n = 250, 500$, and 1000, in which a sink node is located at the center of this field (Fig. 4). We simulate all the sensed values collected to the sink. Here, the average distance between neighbor nodes is 31.6[m] with $n = 250$, 22.4[m] with $n = 500$, and 15.8[m] with $n = 1000$. Note that this setting is designed under the assumption of 10[dBm] transmission power with 100[m] communication range, which is the legal transmission power limitation in Japan. If we adopt 20[dBm] case, which is the major limitation in Europe, and 30[dBm], which in US, the physical scale will be expanded by 3.16 $times$ and 9.99 $times$, respectively, in theoretical calculation. In reference, we roughly estimated the WuR range of each transmission power from several survey articles such as [3] and numerical calculation, which is 15[m], 44[m], and 140[m] for 10, 20, and 30[dBm] cases in 2.4GHz band, respectively. Although the conditions and hardware of reference information on WuR vary in wide range, we conclude that WuR-based methods hardly run the scenario in practice.

The simulation parameters we used are shown in Table I. We assume normal AA battery (×2) to drive sensor devices, and we use the general beacon and sensing intervals that we think are appropreate for general environmental sensing cases. Beacon, Data, and Ack frame size is computed based on IEEE 802.15.4 frame format plus required field size specific to JBS. Back-off window is also determined based on IEEE 802.15.4 specifications. Value t_{txwait} represents the mean

TABLE I
SIMULATION PARAMETERS

Variable	Exprain	Value
C_{batt}	Battery Capacity	2500[mAh]
V	Operate Voltage	3[V]
b	Beacon Interval	30,60[sec]
r	Sensing Interval	10,30,60[min]
l_{beacon}	Beacon Frame Size	37[Bytes]
l_{data}	Data Frame Size	43[Bytes]
l_{ack}	Ack Frame Size	17[Bytes]
t_{txwait}	Listening Time for Beacons	100[msec]
$t_{backoff}$	Backoff Window	2.56[msec]
n	# of Nodes	250,500,1000
C_{slimit}	Min. Power as *Relay*	700[mAh]
C_{change}	Power to Trigger Repair	500[mAh]

waiting time of senders to wait for beacons after waking up, which is an important value in estimating energy consumption. Although senders can learn the timing of receiver's periodical beacons, we have to consider time drift of sensor devices. For the value $t_{txwait} = 100[msec]$, we assume general inexpensive crystal real-time clock(RTC) module and computed the required waiting time for the clock precision and sensing interval (i.e., wake-up interval 60[min]). 100[msec] is about the twise required time to absorb the time drift. Two threshold values C_{slimit} and C_{change} was determined to optimize the performance of JBS in this scenario.

As for the criteria for performance measurement, we use the lifetime of the network, which is measured as the time duration from when all nodes first transit to *stable* state until the network dies. We regard that the network dies when we have a node in the network that runs out of residual power, or when the delivery tree is divided due to failure of local reconstruction. (Since JBS do not use low-power nodes as *relay* nodes, delivery tree may be divided even if all nodes are alive.) We performed 5 simulation executions with different random seeds for each set of parameter values, and used the average to evaluate the performance.

C. Hardware Models and Settings

A large number of sensor-node prototypes have been developed so far in both academic and industrial teams, but they in fact use a small variation of microcontrollers (MCUs) and RF-chips. Because the characteristics in power consumption performance is mostly dominated by the specifications of those chips, we can model the up-to-date power-consumption characteristics as the combinations of chips used in the sensor devices. In our simulation, we use the hardware model structured as a typical excellent low-power sensor-network device to estimate the real performance of JBS with the current devices.

A list of recent prototype devices are available on web sites such as [18]. In many cases, sensor nodes that work on 2.4GHz band adopts Chipcon CC2420 [19] developed by Texas Instruments as a RF controller, and MSP430F2617 [20] or MSP430F1611 [21] as MCU. Accordingly, as the typical equipment that has the best energy consumption performance, we adopt CC2420 and MSP430F2617 in our hardware model.

TABLE II
SPECIFICATIONS ON POWER CONSUMPTION

Operation	Time[sec]		Current[mA]	
CC2420 RF Controller				
Data TX (1byte)	t_{txb}	416×10^{6}	I_{txb}	17.4
Data RX (1byte)	t_{rxb}	416×10^{6}	I_{rxb}	19.7
Wakeup	t_{Rwake}	1.35×10^{3}	I_{Rwake}	0.426
Idle Listening		-	I_{Rwait}	0.426
Sleep		-	I_{Rsleep}	0.02
MSP430F2617 Microcontroller				
Operating	-	-	I_{active}	4.12
Sleep	-	-	I_{sleep}	1.10×10^{3}
Wakeup	t_{wake}	1.0×10^{6}	I_{wake}	4.12
ADT7410 Temparture Sensor				
Sensing	t_{sense}	2.4×10^{1}	I_{sense}	2.1×10^{1}

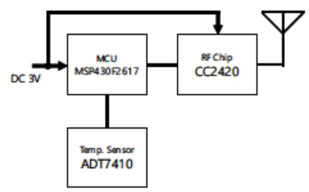


Fig. 5. Sensor Node Block Diagram

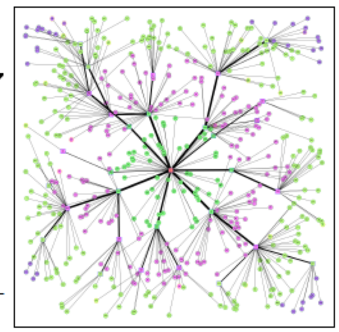


Fig. 6. A Delivery Tree Build by JBS $n = 500$)

We persent the overview of MSP430F2617 for reference: The MSP430F2617 is an ultra-low power 16-bit MCU supplied by Texas Instruments. It operates on 1.8-3.6[V], and the current in operation on 2.2[V] and 1[MHz] is 365[μA], in sleeping state is 0.5[μA], and in power off state is 0.1[μA]. It can recover from sleep state within 1 [msec]. There is a sensor node that use both MSP430F2617 and CC2420, which is Z1 [23] developed by Zolertia.

The block diagram of our hardware model is shown in Fig. 5, which connects MCU (MSP430F2617), RF Controller (CC2420), and temperature sensor (ADT7410 [22]). The official specifications on power consumption is shown in Table II, from which we compute the parameters related with powers. Because basically the power consumption of duty-cycled sensor devices is dominated by those controllers and sensors, the hardware model leads us to the precise estimation of the lifetime of latest sensor-network devices.

D. The Energy Consumption Model

In the simulation, we accumulate the energy consumption based on an energy consumption model. Energy consumption models has been often used in the literature to estimate the lifetime of sensor nodes in simulation. We also define an energy consumption model similar to the existing ones, but being aware of the hardware model described in the previous section. In the following, we clarify the power consumption model and the way to estimate the power consumption based on the hardware model in our evaluation.

We present the energy consumption model as follows.

$$E = E_{rx} + E_{tx} + E_{listen} + E_{wake} + E_{sleep} + E_{sense}, \quad (1)$$

where E is the total power, E_{rx} is the reception power, E_{tx} is the transmission power, E_{listen} is the idle listening power, E_{wake} is the power required to wake up from sleep, E_{sleep} is the power consumed in sleep mode, and E_{sense} is the power consumed to obtain values from the sensor.

In our simulation, we accumulate the power consumption of each items in formula (1) using the variables used in the formula are shown in Table I and II. We present the exression of E_{tx} as follows. Other itmes are expressed similarly according the the protocol behavior.

E_{tx}: Transmission power consists of the power to transmit beacon, data, and ack frames. The power for beacon frames is expressed by $Vl_{beacon}t_{txb}(I_{txb} + I_{active})$. Similarly, we have $Vl_{data}t_{txb}(I_{txb}+I_{active})$ for data frames and $Vl_{ack}t_{txb}(I_{txb}+I_{active})$ for ack frames. Consequently, E_{tx} is expressed as the following formula (2).

$$\begin{aligned} E_{tx} &= NUM_{Btx}Vl_{beacon}t_{txb}(I_{txb} + I_{active}) \\ &\quad +NUM_{Dtx}Vl_{data}t_{txb}(I_{txb} + I_{active}) \\ &\quad +NUM_{Atx}Vl_{ack}t_{txb}(I_{txb} + I_{active}), \quad (2) \end{aligned}$$

where NUM_{Btx}, NUM_{Dtx}, and NUM_{Atx} are the number of transmissions of beacons, data, and ack frames obtained in the simulation.

E. Evaluation Results

1) Lifetime of Networks: We run simulations for both JBS and RI-MAC and obtained the results on lifetime. In JBS, we observed that several local reconstruction processes work and in most of the cases they finished quickly without expanding the reconstruction area by making additional children *relay* nodes transit to *construction* state. For reference, we illustrate the delivery tree build by JBS in Fig. 6.

We present the result on network lifetime. Fig. 7 presents the comparison of network lifetime between JBS and RI-MAC (with load-balanced routing). In RI-MAC, the lifetime is about 3 years in case of 250 nodes, and reduces as the number of nodes increases to 500 and 1000. This is because of the increased load of data traffic generated by a large number of nodes. In contrast, JBS lives about 6-years in case of 250 nodes, which outperforms RI-MAC by twice. This improvement is achieved by saving the energy at leaf nodes and reusing the saved energy as relay nodes after tree reconstruction. Also note that JBS keeps the lifetime even though the number of nodes increase. This is because the total energy preserved in the network is increased when the number of nodes increases. JBS succeded to utilize the potential energy by adaptively reconstructing delivery tree.

2) Performance with Various Parameter Values: Next, we see the effect of parameter values that can be modified by users. As the important parameters that would have operationally essential and simultaneously would have large effect on lifetime, we examine the variation of beacon interval b and sensing interval r.

Fig. 8 shows the lifetime of JBS where sensing interval is fixed as $r = 30$[min] and beacon intervals vary as $b = 30$ and 60[sec]. It is natural that the case $b = 60$ has longer lifetime than the case $b = 30$, and the difference reflects on the load of beacon transmission. Note that the difference shrinks when the number of nodes increases. This means that the significance of data transmission load compared to beacons transmission load is larger in $b = 60$[sec] case than $b = 30$[sec] case.

Fig. 9 and 10 show the results in which beacon interval is fixed as 30[sec] and 60[sec], respectively, while sensing interval is varied as $r = 10, 30$ and 60[min]. In both beacon intervals $b = 30$ and 60[sec], lifetime is longer when sensing interval is larger. However, the difference in performance is not so significant. This is because the sensing interval is far larger than beacon intervals in time so that the impact of sensing interval on lifetime is relatively smaller than beacon interval.

3) Residual Power Distribution: In Fig. 11 and 12, we depict the distribution of residual power in case of $n = 500$ when the network dies. We see that, in case of JBS shown in Fig. 11, most nodes have more than 1250[mAh] battery power, whereas in case of RI-MAC, shown in Fig. 12, residual battery power distributes uniformly at low level. It is indicated that JBS still preserves total energy in the network. The nodes with large residual power were *leaf* nodes, and only the small number of *relay* nodes are fully run out of battery power. JBS finishes its lifetime because the nodes located near the sink has run out of battery power, where the load of data traffic is the most concentrated.

Note that this is the favorable property of JBS in real operation of sensor networks because the network lifetime is able to prolong if we reinforce the battery power of the bottleneck part of the network. Or, it is said that we can recover the network by replacing the battery of only a part of the network. JBS has a favorable property that nodes are clearly classified into two parts, i.e., nodes that run out of power and those have enough residual power, meaning that we can operate the sensor network by replacing batteries of the former part repeatedly, which potentially will lead low maintenance cost in practice.

V. Conclusion

In this paper, we proposed a new approach of joint MAC and routing scheme based on receiver-initiated MAC protocols for duty-cycled wireless sensor networks. Based on the approach, we present the specific design of the protocol called JBS. In JBS, we save power consumption by omitting periodical beacons on the leaf nodes of delivery tree. The routing protocol build a delivery tree such that the number of relay node is minimized to save as much power consumption as possible. To enable adaptive routing against topology changes, a part of leaf

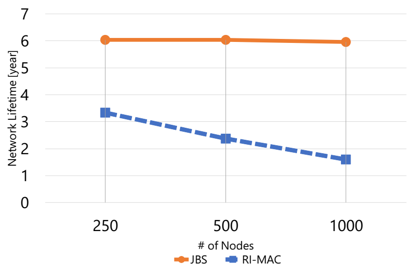


Fig. 7. Lifetime of JBS and RI-MAC

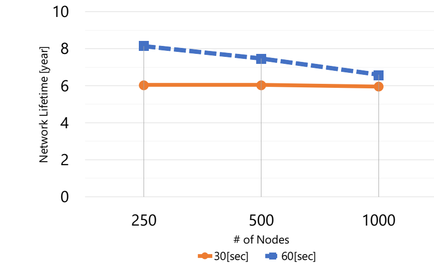


Fig. 8. Lifetime with Fixed Sensing Interval $r =$ 30[min], Varied Beacon Interval $b = 30, 60$[sec]

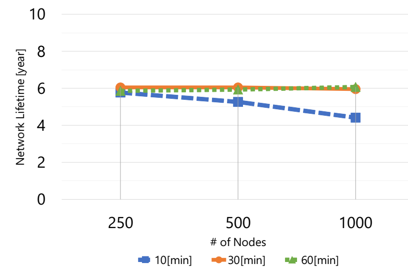


Fig. 9. Lifetime with Fixed Beacon Interval $b = 30$[sec], Varied Sensing Interval $r = 10, 30, 60$[min]

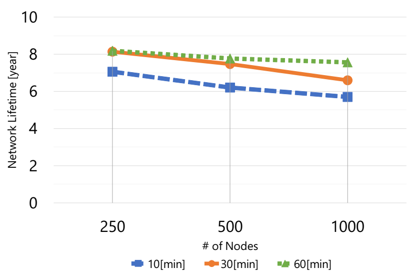


Fig. 10. Lifetime with Fixed Beacon Interval $b =$ 60[sec], Varied Sensing Interval $r = 10, 30, 60$

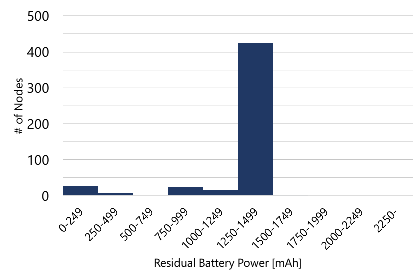


Fig. 11. Residual Battery Power of JBS When Lifetime Ends ($n = 500$).

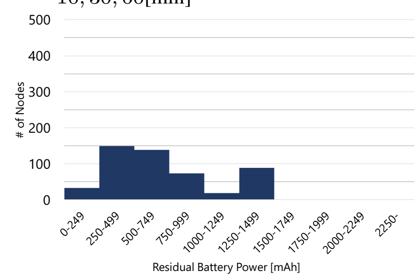


Fig. 12. Residual Battery Power of RI-MAC When Lifetime Ends ($n = 500$).

nodes becomes so called researved relay nodes that provides help for local communications to find detour paths efficiently. By adaptively reconstructing the delivery tree in face of low-residual-power events, JBS prolongs the network lifetime. Through simulation of a environmental sensing scenario with several hundreds of nodes, we confirmed that JBS achieves more than twice of lifetime compared with the conventional methodology. As a result, we showed that conbination of receiver-initiated MAC and routing functionality could draw a new potential of energy saving mechanism in duty-cycled wireless sensor networks.

References

[1] Z. Li, Y. Peng, D. Qiao, and W. Zhang, "Joint Aggregation and MAC design to prolong sensor network lifetime," International Comference on Network Protocol (ICNP'13), 2013.

[2] A. Bachir, M. Dohler, T. Watteyne, and K.K. Leung, "MAC Essentials for Wireless Sensor Networks," IEEE Communications Surveys & Tutorials 12(2), 2010.

[3] F.Z. Djiroun, and D. Djenouri, "MAC Protocols with Wake-up Radio for Wireless Sensor Networks: A Review," IEEE Communications Surveys & Tutorials, 19(1), 2017.

[4] J. Oller, I. Demirkol, J. Casademomt, J. Paradells, G.U. Gamm, and L. Reindl, "Has Time Come to Switch from Duty-cycled MAC Protocols to Wake-up Radio for Wireless Sensor Networks?," IEEE/ACM Transactions on Networking, 24(2), 2016.

[5] V. Rodoplu and T.H. Meng, "Minimum Energy Mobile Wireless Networks," IEEE Journal of Selected Areas of Communications, 17(8):1333-1344, 1999.

[6] K. Langendoen, "The MAC Alphabet Soup," http://www.st.ewi.tudelft.nl/koen/MACsoup/protocols.php

[7] IEEE Computer Society: *IEEE Std 802.15.4-2003, IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks Specific requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer* (2003).

[8] X. Zhang and Z.D. Wu, *The balance of routing energy consumption in wireless sensor networks,* Journal of Parallel and Distributed Computing, Vol.71 Issue 7, pp.1024–1033 (2011).

[9] X.H. Li, K.L. Fang, H.P. Chen, H.S. Ho, *Energy-balance routing for wireless sensor networks with scale-free characteristic*, Telecommunication Networks and Applications Conference (ATNAC) (2012).

[10] Y. Sum, O. Gurewits, and D. B. Johnson: *RI-MAC: A Receiver-initiated Asynchronous Duty Cycle MAC Protocol for Dynamic Traffic Loads in Wireless Sensor Networks*, In Proc. of SenSys '08, pp.1–14 (2008).

[11] J. Polastre, J. Hill, and D. Culler, *Versatile Low Power Media Access for Wireless Sensor Networks*, In Proc. of SenSys '04, pp.95–107 (2004).

[12] M. Buettner, G. Yee, E. Anderson, and R. Han: *X-MAC: A Short Preamble MAC Protocol for Duty-cycled Wireless Sensor Networks*, In Proc. of SenSys '06 (2006).

[13] P. Huang, C. Wang, L. Xiao: *RC-MAC: A Receiver-Centric MAC Protocol for Event-Driven Wireless Sensor Networks*, In Proc. of IWQoS '10 (2010).

[14] P. Dutta, S. Dawson-Haggerty, Y. Chen, C.M. Liang, and A. Terzis, "A-MAC: A Versatile and Efficient Receiver-Initiated Link Layer for Low-Power Wireless," ACM Trans. on Sensor Networks, 8(4), 2012.

[15] P. Gupta and P. Kumar, "The Capacity of Wireless Networks," IEEE Transaction on Information Theory, Vol.46, No.2, pp.388–404 (2000).

[16] D. Luo, X. Zhu, X. Wu, and G. Chen, *Maximizing Lifetime for the Shortest Aggregation Tree in Wireless Sensor Networks*, In Proc. of INFOCOM '11 (2011).

[17] T.W. Kuo and M.J. Tsai, *On the Construction of Data Aggregation Tree with Minimum Energy Cost in Wireless Sensor Networks: NP-Completeness and Approximation Algorithms*, In Proc. of INFOCOM '12 (2012).

[18] List of Wireless Sensor Nodes (Wikipedia), https://en.wikipedia.org/wiki /List_of_wireless_sensor_nodes

[19] Texas Instruments Inc.: *CC2420 Data Sheet*, http://www.ti.com/lit/ds/symlink/cc2420.pdf

[20] Texas Instruments Inc.: *MSP430F2617 Data Sheet*, http://www.ti.com/lit/ds/symlink/msp430f2617.pdf

[21] Texas Instruments Inc.: *MSP430F1611 Data Sheet*, http://www.ti.com/lit/ds/slas368g/slas368g.pdf

[22] Analog Devices Inc.: *ADT7410 Data Sheet*, http://www.analog.com/ media/en/technical-documentation/data-sheets/ADT7410.pdf

[23] Zolertia Inc.: *Z1*, http://zolertia.io/z1