I^{2}C: A Holistic Approach to Prolong the Sensor Network Lifetime

Yang Peng, Zi Li, Daji Qiao, and Wensheng Zhang
Iowa State University, Ames, IA 50011
{yangpeng, zili, daji, wzhang}@iastate.edu

***Abstract*—We present a novel holistic approach (called I^{2}C – Intra-route and Inter-route Coordination) to prolong the sensor network lifetime under the end-to-end delivery delay constraint. I^{2}C is composed of two lifetime balancing modules: (i) the Intra-Route Coordination module that allows the nodes on the same route to balance their nodal lifetimes through adjusting the MAC behaviors collaboratively; (ii) the Inter-Route Coordination module that balances the nodal lifetimes across different routes via adjusting the communication routes. Different from existing works which conduct either intra-route or inter-route lifetime balancing, or a simple combination of the two, I^{2}C leverages the advantages of both techniques with a sophisticated design that emphasizes the awareness and collaboration between two modules. Thus, I^{2}C is able to prolong the network lifetime much more effectively than the state-of-the-art solutions, while guaranteeing the desired delay bound and maintaining a similar level of network power consumption. This has been demonstrated with extensive ns-2 simulation and TinyOS experiment results.**

I. Introduction

When applying sensor networks for long-term applications such as continuous monitoring, how to prolong the network lifetime is of critical importance. For these applications, *network lifetime* is often defined as the minimal nodal lifetime among all nodes in the network [1]–[3]. In addition to operating sensor nodes at a low duty cycle to conserve energy, many works have been proposed to approach this goal via balancing the distribution of nodal lifetime in the network.

A. Motivations

Energy-aware routing and *intra-route coordination* are two nodal lifetime balancing techniques commonly used in sensor networks to prolong the network lifetime. The energy-aware routing schemes [4], [5] attempt to balance the nodal lifetime through distributing more communication workload to routes that contain nodes with longer nodal lifetime and/or higher residual energy. However, as these schemes balance the nodal lifetime through re-routing only, bottleneck nodes such as the nodes close to the sink may still consume more energy than others in the network and thus bound the network lifetime.

Different from energy-aware routing, the intra-route coordination schemes [6]–[8] attempt to balance the nodal lifetime of nodes along the same routing path such that the communication workload at the bottleneck nodes can be shifted to other nodes on the same route but with a higher nodal lifetime. Though intra-route coordination can overcome the bottleneck effects efficiently, it may not fully utilize the network energy resources, as it only attempts to balance nodal lifetime within a route but cannot balance the nodal lifetime of nodes belonging to different routes.

Therefore, it is necessary and beneficial to have an integrated scheme which can take advantage of both energy-aware routing and intra-route coordination and meanwhile avoid their limitations. However, without careful analysis and design, simply operating existing energy-aware routing and intra-route coordination schemes together may not provide an efficient solution. For example, as shown in Figure 1, the network lifetime achieved by a simple combination of energy-aware routing and intra-route coordination is comparable to that achieved by intra-route coordination alone.

number of nodes	IaC	EA+IaC	I^{2}C
25	47.1h	43h	60.2h
100	16.5h	18.5h	25.6h

Fig. 1. Network lifetime comparison between intra-route coordination only (denoted as IaC), a simple combination of energy-aware routing and IaC (denoted as EA+IaC), and our proposed I^{2}C schemes. The data generation interval is 40 seconds and the number of nodes in the network varies from 25 to 100. These results are extracted from our ns2-based simulation results in Section V.

B. Contributions

To remedy the deficiencies of either energy-aware routing or intra-route coordination, or a simple combination of the two, we propose a novel holistic approach in this paper, called I^{2}C (*Intra-route and Inter-route Coordination*), which leverages the two lifetime balancing techniques.

The proposed I^{2}C scheme is composed of two core modules: Intra-Route Coordination and Inter-Route Coordination, which are designed to work together in a collaborative manner. For example, with I^{2}C, the new parent node of a sensor node may not simply be the one with the highest nodal lifetime (among all potential parent nodes). Rather, it is the one with the maximal potential to increase the minimal nodal lifetime among the node's neighborhood. I^{2}C accomplishes this by predicting the nodal lifetimes after the potential route switch, via close collaboration between the two modules. Due to such a sophisticated design, I^{2}C is able to prolong the network lifetime more effectively and efficiently, as shown in Figure 1. The contributions of this work are summarized below.

- To the best of our knowledge, I^{2}C is the first holistic approach which leverages both inter-route (i.e., energy-aware routing) and intra-route lifetime balancing techniques for duty cycle sensor networks.
- I^{2}C is a distributed and lightweight solution. It works through limited control information exchange locally between neighbor nodes.

978-1-4673-5946-7/13/$31.00

- I^{2}C has been implemented and evaluated, and it achieves significant improvement on network lifetime over the state-of-the-art solutions.

C. Organization

The rest of the paper is organized as follows. Section II summarizes the related work. Section III presents the system model, the theoretical analysis, the problem statement, and an overview of the proposed I^{2}C scheme. Details of the I^{2}C design are described in Section IV. Section V shows the performance evaluation results obtained from both ns-2 simulations and TinyOS testbed experiments. Finally, Section VI concludes the paper.

II. Related Work

Among the techniques to prolong the network lifetime, multiple energy-aware routing protocols have been proposed for ad hoc and sensor networks and [3]–[5] are representative ones among them. Recently, authors in [2], [9], [10] proposed specially-designed energy-aware routing schemes for duty cycle sensor networks. In all these works, the main idea is to route packets through nodes with a higher residual energy or a longer nodal lifetime such that nodes with a lower energy or a shorter lifetime can participate less in data transmission activities. As a result, the minimum nodal lifetime in the network may be extended and the network lifetime may be prolonged.

Intra-route lifetime balancing, as another approach to prolong the network lifetime, has also been studied in [6]–[8], [11], [12]. Particularly, SEESAW [6] was proposed to balance the energy consumption between sender and receiver through adapting the data retry interval at the sender side and the channel checking period at the receiver side. ZeroCal [11] targets at improving the fairness of energy utilization in duty cycle sensor networks by dynamically tuning the nodal wakeup interval. Different from ZeroCal, GDSIC [7] decides the individual nodal wakeup interval through solving distributed convex optimization problems. Though the network lifetime can be prolonged by these schemes, they do not guarantee the end-to-end delay bound. pTunes [12] is a recently proposed centralized solution which adjusts the MAC parameters dynamically for low-power sensor networks. It formalizes a multi-objective optimization problem, in which prolonging network lifetime and guaranteeing the end-to-end delay can be solved together.

In addition to the inter-route and intra-route lifetime balancing schemes, approaches to prolong the network lifetime through cross layer design are proposed in [13]–[19]. In these works, [15] attempts to maximize the network lifetime via joint routing and MAC design, [19] solves the problem via joint routing and congestion control, and [16] tackles the problem through joint optimal design of physical, MAC, and routing layers in time slotted networks. Though these works can prolong the network lifetime, they either impose high overhead to the system or are not designed in a collaborative manner. More importantly, most of these works are not suitable for duty cycle sensor networks.

III. System Model and Design Overview

A. System Model

We study the problem of prolonging the network lifetime of a sensor network that is configured for long-term monitoring applications. Each node in the network generates and reports sensory data periodically and all nodes form a data collection tree rooted at the sink. The data collection tree is maintained and updated through periodic routing update messages exchanged between neighbor nodes. We do not assume data aggregation in this work.

At the MAC layer, the design principle of our proposed scheme does not require a particular MAC protocol underneath the routing layer. In fact, it works fine with other duty cycle MAC protocols as well, as long as the node's MAC behavior and duty cycle are adjustable [6], [8], [20]. In this work, to simplify the presentation, we assume that each node runs an RI-MAC [20] like protocol as follows. As shown in Figure 2, in order to receive a data packet, a node wakes up every T_r interval to interact with potential senders. Upon wakeup, it sends out a beacon and then checks the channel activity for ϕ time for incoming data packets. If a data packet is received within ϕ time, it replies with an ACK; otherwise, it goes back to sleep. On the other hand, if a node has a packet to send, it remains awake and waits idly for the target receiver's beacon to start the data transmission (with a duration of τ). Different from the RI-MAC protocol which has a fixed T_r, we assume that T_r is a tunable MAC layer parameter in this work.

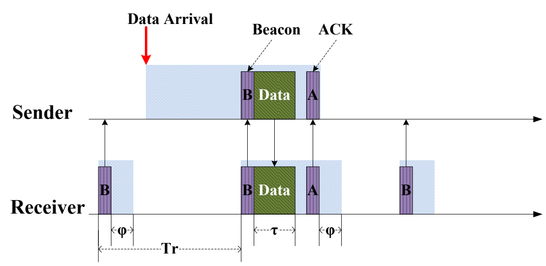


Fig. 2. An RI-MAC like protocol but with a tunable T_r parameter.

B. Nodal Lifetime

With the MAC protocol described in the previous section, the nodal lifetime of node i can be estimated as follows:

$$L(i) = \frac{e(i)}{c(i)}, \quad (1)$$

where $e(i)$ is the residual energy and $c(i)$ is the energy consumption rate:

$$c(i) = \sum_{j \in \Omega(i)} f(i,j) \left(\tau + \frac{T_r(j)}{2} \right) P + \sum_{k \in \Omega(i)} f(k,i) \tau P + \frac{\phi(i)}{T_r(i)} P. \quad (2)$$

Here, $\Omega(i)$ is the set of i's neighbor nodes, $f(i,j)$ is the traffic rate from i to j, and P is the amount of energy consumed when the node's radio is on for one unit of time.

In the above estimation, the short beacon and ACK transmissions are omitted. Therefore, to send a data packet to j, i needs to wait for $\frac{T_r(j)}{2}$ time on average, and the data transmission duration is τ. As a result, it consumes

$\sum_{j \in C(i)} f(i,j)\left(\tau + \frac{T_r(j)}{2}\right) P$ power on average for data transmissions. Similarly, the second term in Equation (2) represents the average power consumed for data receptions, and the third term is the average power consumed for monitoring the channel activity for ϕ time every T_r interval.

From Equations (1) and (2), it is interesting to see that nodal lifetime of node i is affected by two factors: (i) the routing behaviors of sensor nodes which decide the outgoing and incoming data rates to i, i.e., $f(i,j)$ and $f(k,i)$; and (ii) the T_r values of i and its receivers, i.e., their MAC behaviors.

C. End-to-End Delivery Delay

With the MAC protocol described in Section III-A, the worst-case one-hop packet delivery delay from i to j is simply

$$D_{i \to j} = T_r(j). \tag{3}$$

Subsequently, the worst-case end-to-end packet delivery delay from a source node to the sink node is

$$\mathcal{D}_{\text{src}\to\text{sink}} = \sum_{\texttt{all hops from source to sink}} D_{i \to j}. \tag{4}$$

From Equation (4), we can see that, similar to nodal lifetime, the end-to-end packet delivery delay is also affected by two factors: (i) the routing behaviors of sensor nodes which decide the route from source to sink; and (ii) the MAC behaviors of sensor nodes which decide the T_r values.

D. Problem Statement

From the above analysis, it is clear that, in order to effectively prolong the network lifetime of a sensor network under the end-to-end packet delivery delay constraint, it is critical to have a holistic approach that adjusts both routing and MAC behaviors of sensor nodes together, which is precisely the goal of this work. Formally, it can be described as follows:

Given:

- For each node i, its residual energy $e(i)$, data generation rate $\lambda(i)$, and set of neighbor nodes $\Omega(i)$.

Objective:

- $\max \min L(i)$, where $L(i)$ is the nodal lifetime of i and can be calculated using Equation (1).

Subject to:

- *Network Flow Constraint:* for each sensor node i, $\sum_{k \in \Omega(i)} f(k,i) + \lambda(i) = \sum_{j \in \Omega(i)} f(i,j)$.
- *End-to-End Delay Requirement:* $\mathcal{D}_{\text{src}\to\text{sink}} \leqslant \mathcal{D}_{\text{e2e}}$ for all source nodes, where $\mathcal{D}_{\text{e2e}}$ is an application-specified delay bound.[1]
- $\forall i, j,\ f(i,j) \geqslant 0$.
- $\forall i,\ T_r(i) > 0$.

Output:

- For each node i in the network, its MAC behavior, i.e., $T_r(i)$, and its routing behavior, i.e., $f(i,j),\ \forall j \in \Omega(i)$.

[1]This value can be determined before deployment, or dynamically updated after deployment. In the latter case, the update can be disseminated through sink-to-node communications [21], [22] or piggybacked in a packet and disseminated hop by hop.

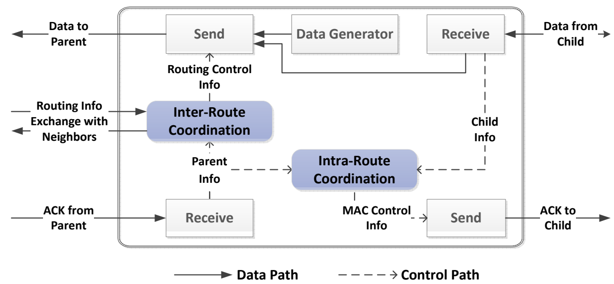


Fig. 3. Overview of the proposed I^{2}C scheme.

E. Design Overview

Directly solving the above optimization problem by individual nodes is impractical because it requires each node to collect the following information from every other node in the network: residual nodal energy, data generation rate, and network topology. Acquiring these information could incur very high communication overhead because of potentially large network scale and dynamic nature of the information. So instead, we propose a distributed, localized, and low-cost solution, called I^{2}C (Intra-route and Inter-route Coordination).

In I^{2}C, coordinations only take place between neighbor nodes which exchange lightweight control information and adjust their routing and MAC behaviors together in a collaborative manner. As shown in Figure 3, when a parent node receives a data packet from its child node, it extracts the control information (e.g., the expected nodal lifetime) embedded in the data packet and feeds them into the *Intra-Route Coordination* module, which decides how the node shall adjust its MAC behavior (i.e., T_r). It also decides how the child node shall adjust its T_r and piggybacks the decision into the ACK packet to the child node, based on which the child node adjusts its MAC behavior accordingly. This way, the shorter nodal lifetime between parent and child nodes can be extended (at the expense of the other one). Moreover, a child node may also decide (via the *Inter-Route Coordination* module) to adjust its routing behavior by selecting a different parent node for future communications. With such inter-route coordination, the network lifetime may be extended further as the overall network resource may be utilized more efficiently. For example, the minimal nodal lifetime between the child node, the current parent node, and the new parent node may be extended more (at the expense of the other two). Both coordination modules operate under the condition that the end-to-end delay requirement shall be satisfied. Details of the modules will be elaborated in Sections IV-A and IV-B.

IV. The Proposed I^{2}C Scheme

In this section, we describe the details of the two core modules of the proposed I^{2}C scheme: *Intra-Route Coordination* and *Inter-Route Coordination*.

A. Intra-Route Coordination

The Intra-Route Coordination module coordinates between neighbor nodes on the same route of the current data collection tree. More specifically, it coordinates the MAC behaviors of a pair of parent-child nodes, and adjusts their MAC parameters (i.e., T_r) in a collaborative manner whenever there are data communications between them. I^{2}C achieves this goal by piggybacking lightweight control information in the data/ACK exchanged between parent-child nodes.

1) Parent Node's Behavior: Every T_r interval, a parent node j in the data collection tree turns on radio, sends a beacon, and monitors the channel for ϕ time. During the monitoring period, if a data packet is received from a child node i, the following information will be extracted from the data packet:

- $L(i)$ – *i's estimated nodal lifetime;*
- $T_r(i)$ – *i's MAC parameter;*
- $\mathcal{D}_{leaf\to i}$ – *the maximal delivery delay from the leaf nodes on the data collection subtree rooted at node i to node i.*

By comparing $L(i)$ with its own nodal lifetime $L(j)$, node j attempts to adjust its T_r differently in the two cases discussed below, and then embeds the updated T_r (denoted as T_r^{new}) in the ACK to node i. Note that, according to Equations (1) and (4), the adjustment of T_r not only affects the nodal lifetime of both parent and child nodes, but the end-to-end delivery delay as well. Therefore, j needs to make sure that the following conditions are satisfied after the T_r adjustment:

$$\begin{cases} \max\limits_{i\in\Phi(j)} \mathcal{D}_{\text{leaf}\to i} + T_r^{\text{new}}(j) + \mathcal{D}_{j\to\text{sink}} \leqslant \mathcal{D}_{\text{e2e}}, \\ T_r^{\text{new}}(i) > 0. \end{cases} \tag{5}$$

Here, $\Phi(j)$ is the set of j's children nodes, and $\mathcal{D}_{j\to\text{sink}}$ is the delivery delay from j to the sink, which is maintained locally by j and also embedded in the ACK to i.

Case 1: $L(j) > L(i)$**.** In this case, j decreases $T_r(j)$ by a small amount[2]. Correspondingly, i will increase $T_r(i)$ by the same small amount. This procedure repeats every time when a data packet is received, till $T_r(j)$ reaches a default minimal value (which is used to prevent excessive beacon transmissions that may cause severe channel contention). This way, according to Equation (1), the time that i waits before transmitting a data packet to j is reduced. As a result, i reduces its energy consumption and consequently increases its nodal lifetime, which is at the expense of node j spending more time on periodic channel checking. Note that, as $T_r^{\text{new}}(j) < T_r(j)$ and $T_r^{\text{new}}(i) > T_r(i)$, both conditions in Equation (5) are satisfied after the T_r adjustment.

Case 2: $L(j) < L(i)$**.** In this case, j may increase $T_r(j)$ to reduce its energy consumption for idle listening and increase its nodal lifetime, as long as the conditions in Equation (5) are satisfied. This can be guaranteed if $T_r^{\text{new}}(j)$ satisfies:

$$\begin{cases} \mathcal{D}_{j\to\text{sink}} + T_r^{\text{new}}(j) + \max\limits_{y\in\Phi(i)} \mathcal{D}_{\text{leaf}\to y} < \mathcal{D}_{\text{e2e}}, \\ \mathcal{D}_{j\to\text{sink}} + T_r^{\text{new}}(j) + \max\limits_{x\in\Phi(j)-i} \mathcal{D}_{\text{leaf}\to x} \leqslant \mathcal{D}_{\text{e2e}}. \end{cases} \tag{6}$$

This is because such $T_r^{\text{new}}(j)$ can always be accommodated by decreasing $T_r(i)$ to:

$$T_r^{\text{new}}(i) = \mathcal{D}_{\text{e2e}} - \mathcal{D}_{j\to\text{sink}} - T_r^{\text{new}}(j) - \max_{y\in\Phi(i)} \mathcal{D}_{\text{leaf}\to y}, \tag{7}$$

[2]In our implementation, we adjust T_r by 20 ms each time. The reason for choosing a small adjustment step is to avoid the potential thrashing effect that may be caused by the following factors: (i) the nodal lifetime estimation may be inaccurate; (ii) multiple nodes may adjust T_r simultaneously; and (iii) the data collection tree varies over time as nodes may join and leave at any time.

since we have $T_r^{\text{new}}(i) > 0$ by plugging the first condition in Equation (6) into Equation (7), and

$$\begin{aligned} & \mathcal{D}_{\text{leaf}\to i}^{\text{new}} + T_r^{\text{new}}(j) + \mathcal{D}_{j\to\text{sink}} \\ = & \max_{y\in\Phi(i)} \mathcal{D}_{\text{leaf}\to y} + T_r^{\text{new}}(i) + T_r^{\text{new}}(j) + \mathcal{D}_{j\to\text{sink}} \\ = & \mathcal{D}_{\text{e2e}}. \end{aligned} \tag{8}$$

Combining Equation (8) with the second condition in Equation (6), we can see that the end-to-end delivery delay requirement is guaranteed after the T_r adjustment.

Table I gives three examples to illustrate the parent node's behavior, where the first example corresponds to Case 1, and the second and third examples correspond to Case 2. Take the third example for instance. The parent node j intends to increase its T_r by $20ms$ since $L(j) = 20h < 30h = L(i)$. However, Equation (6) (more specifically, the second condition in Equation (6)) is not satisfied, meaning that the intended increment in $T_r(j)$ would result in a violation of the end-to-end delay requirement of $20s$. Therefore, j instead sticks with the current T_r till the arrival of the next data packet, which leaves the nodal lifetimes between itself and its child node i temporarily unbalanced.

2) Child Node's Behavior: When a child node i has a data packet to send, it turns on radio and waits idly for its parent node j's beacon to start the data transmission. After an ACK is received for the data packet, it extracts the $T_r^{\text{new}}(j)$ information carried in the ACK and simply adjusts its own T_r to:

$$T_r^{\text{new}}(i) = \mathcal{D}_{\text{e2e}} - \mathcal{D}_{j\to\text{sink}} - T_r^{\text{new}}(j) - \max_{y\in\Phi(i)} \mathcal{D}_{\text{leaf}\to y}. \tag{9}$$

B. Inter-Route Coordination

Complementary to the Intra-Route Coordination module, the Inter-Route Coordination module attempts to extend the network lifetime via dynamic adjustment of the data collection tree. Specifically, based on the control information carried in the routing update messages, each sensor node periodically selects the best neighbor as its parent node towards the sink, which maximizes the minimal nodal lifetime between the node, its current parent, and the new parent. This, essentially, decides how the node's communication workload shall be distributed among neighbors. Different distributions of workload may result in different energy consumption rates and hence different nodal lifetimes among neighbors. As such adjustment is conducted by every node in the network, the nodal lifetimes may be balanced gradually across the entire network.

The goal of inter-route coordination can be formally described as follows. Consider node i in the network. Let j denote its current parent. Let $p_1, \cdots, p_n$ denote the set of i's communication neighbors (excluding j). We denote the lifetimes of these nodes as $L(i)$, $L(j)$, and $L(p_1), \cdots, L(p_n)$, respectively. The goal is to find $p^* \in \{p_1, \cdots, p_n\}$ such that

$$\min(L'(i), L'(j), L'(p^*)) > \min(L(i), L(j), L(p^*)), \tag{10}$$

and

$$\begin{aligned} & \min(L'(i), L'(j), L'(p^*)) \\ = & \max_{p\in\{p_1,\cdots,p_n\}} \min(L'(i), L'(j), L'(p)), \end{aligned} \tag{11}$$

TABLE I
EXAMPLES OF INTRA-ROUTE COORDINATION WITH THE END-TO-END DELAY REQUIREMENT OF 20 SECONDS

child node i with $T_r(i)=1s$		parent node j with $T_r(j)=1s$			T_r adjustment	
$L(i)=20h$	$\mathcal{D}_{\text{leaf}\to i}=10s$	$L(j)=30h$	$\max_{x\in\Phi(j)-i}\mathcal{D}_{\text{leaf}\to x}=10s$	$\mathcal{D}_{j\to\text{sink}}=9s$	$T_r^{\text{new}}(i)=1.02s$	$T_r^{\text{new}}(j)=0.98s$
$L(i)=30h$	$\mathcal{D}_{\text{leaf}\to i}=10s$	$L(j)=20h$	$\max_{x\in\Phi(j)-i}\mathcal{D}_{\text{leaf}\to x}=8s$	$\mathcal{D}_{j\to\text{sink}}=9s$	$T_r^{\text{new}}(i)=0.98s$	$T_r^{\text{new}}(j)=1.02s$
$L(i)=30h$	$\mathcal{D}_{\text{leaf}\to i}=8s$	$L(j)=20h$	$\max_{x\in\Phi(j)-i}\mathcal{D}_{\text{leaf}\to x}=10s$	$\mathcal{D}_{j\to\text{sink}}=9s$	$T_r^{\text{new}}(i)=1s$	$T_r^{\text{new}}(j)=1s$

TABLE II
DECISION MAKING OF THE INTER-ROUTE COORDINATION MODULE

Case	Description			T_r adjustment *if i switches to new parent p* $T_r(i)$	$T_r(p)$	*Reason*
1	$L(p)\leqslant\min(L(i),L(j))$			*Node i shall not switch to new parent p.*		*Switching to p would add more workload to p, thus reducing $L(p)$ and $L_{\min}$.*
2	$L(p)>\min(L(i),L(j))$	$\Delta\mathcal{D}\geqslant 0$	$L(i)<L(p)$	$T_r^{\text{new}}(i)=T_r(i)+\Delta\mathcal{D}$	no change	*Increasing $T_r(i)$ would reduce energy consumed by i for channel checking, which may increase $L(i)$ and $L_{\min}$.*
3			$L(i)\geqslant L(p)$	$T_r^{\text{new}}(i)=T_r(i)+\Delta\mathcal{D}$	no change	*The end-to-end delay requirement prevents $T_r(p)$ from increasing.*
4		$\Delta\mathcal{D}<0$	$L(i)<L(p)$	no change	$T_r^{\text{new}}(p)=T_r(p)+\Delta\mathcal{D}$	*Since p has a longer lifetime, it sacrifices its lifetime to satisfy the end-to-end delay requirement by reducing $T_r(p)$.*
5			$L(i)\geqslant L(p)$	$T_r^{\text{new}}(i)=T_r(i)+\Delta\mathcal{D}$	no change	*Since i has a longer lifetime, it sacrifices its lifetime to satisfy the end-to-end delay requirement by reducing $T_r(p)$.*

where $L'(i)$, $L'(j)$, and $L'(p)$ are the predicted nodal lifetimes of i, j, and p, assuming that (i) node i selects p as its new parent, and (ii) after the route switch, nodes i and p along the new route behave according to the intra-route coordination principle, which are summarized in Table II and details are discussed below. If such p^* can be found, i switches to p^* as its new parent; else, it sticks with the current parent j till the next round of routing update.

To aid the inter-route coordination, each node embeds the following control information in the routing update messages: *nodal residual energy (e)*, *nodal energy consumption rate (c)*, *T_r of node itself and its parent node*, and *delivery delay from the node to the sink ($\mathcal{D}_{\text{node}\to sink}$)*. Based on these information, i can predict the nodal lifetime for each of its potential new parent nodes $p\in\{p_1,\cdots,p_n\}$. As listed in Table II, there are five possible cases.

Case 1: $L(p)\leqslant\min(L(i),L(j))$. Node i shall not choose any neighbor node that belongs to this case. This is because, if i switches to p, more workload would be added to p which will decrease the nodal lifetime of p. Therefore,

$$\begin{aligned}\min(L'(i),L'(j),L'(p))\leqslant L'(p)<L(p)\\ =\min(L(i),L(j),L(p)),\end{aligned}\tag{12}$$

meaning that Condition (10) is not satisfied.

Case 2: $L(p)>L(i)$ (which implies $L(p)>\min(L(i),L(j))$) and $\Delta\mathcal{D}_{\text{leaf}\to i\to p\to\text{sink}}\geqslant 0$, where $\Delta\mathcal{D}_{\text{leaf}\to i\to p\to\text{sink}}=\mathcal{D}_{\text{e2e}}-\mathcal{D}_{\text{leaf}\to i}-T_r(p)-\mathcal{D}_{p\to\text{sink}}$. In this case, if i would select p as its new parent, its future data packets would be relayed towards the sink by p instead of j. Thus, j's nodal lifetime would be increased to:

$$L'(j)=\frac{e(j)}{c(j)-f(i,j)\left(2\tau+\frac{T_r(j\text{'s parent})}{2}\right)P},\tag{13}$$

and p's nodal lifetime would be decreased to:

$$L'(p)=\frac{e(p)}{c(p)+f(i,j)\left(2\tau+\frac{T_r(p\text{'s parent})}{2}\right)P}.\tag{14}$$

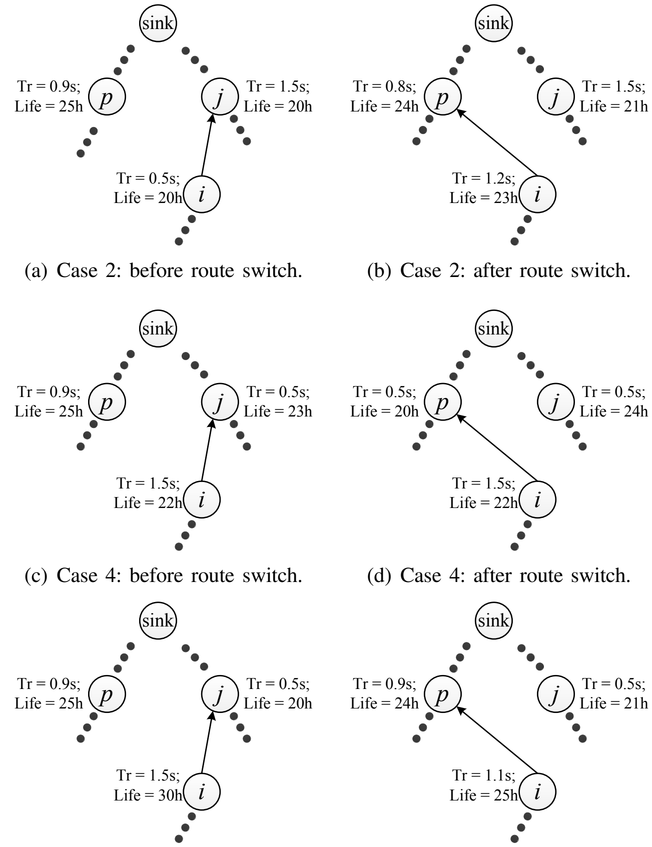
On the other hand, a positive $\Delta\mathcal{D}$ means that i would reach the sink via p with a smaller delay than the required delay bound. This would allow either i or p to increase its T_r (by $\Delta\mathcal{D}$) and consequently the nodal lifetime. As i has a shorter lifetime than p, the intra-route coordination principle would allocate $\Delta\mathcal{D}$ to $T_r(i)$. Therefore, we have

$$L'(i)=\frac{e(i)}{c(i)+\left(f(i,j)\frac{T_r(p)-T_r(j)}{2}-\frac{\Delta\mathcal{D}\cdot\phi}{T_r(i)\cdot(T_r(i)+\Delta\mathcal{D})}\right)P}.\tag{15}$$

An example is given in Figures 4(a) and 4(b). In this example, the minimal nodal lifetime between i, j, and p is increased from 20h to 21h after the route switch. However, in general, as $L'(p)$ and $L'(i)$ depend on many factors, there is no definitive relation between $\min(L(i),L(j),L(p))$ and $\min(L'(i),L'(j),L'(p))$ when $L(p)>\min(L(i),L(j))$ (i.e., Cases 2, 3, 4, and 5). Node i would have to plug in the control information carried in the routing update messages from each potential parent, and check whether Condition (10) is satisfied.

Case 3: $L(i)\geqslant L(p)>L(j)$ (which implies $L(p)>\min(L(i),L(j))$) and $\Delta\mathcal{D}_{\text{leaf}\to i\to p\to\text{sink}}\geqslant 0$. In this case, ideally, p would increase $T_r(p)$ and extend its nodal lifetime. However, as p may have other children nodes, an increase in $T_r(p)$ may result in a violation of the end-to-end delay requirement on other branches of the subtree rooted at p. As a result, we keep $T_r(p)$ unchanged, and allocate $\Delta\mathcal{D}$ to $T_r(i)$ instead. The calculations of the predicated nodal lifetimes are the same as in Case 2.

Case 4: $L(p)>L(i)$ and $\Delta\mathcal{D}_{\text{leaf}\to i\to p\to\text{sink}}<0$. A negative $\Delta\mathcal{D}$ means that the new route via p towards the sink would incur a higher delay than the desired delay bound. In order to reduce the end-to-end delay to be under the bound, $\Delta\mathcal{D}$ has to be absorbed by either i or p. In this case, as p has a longer nodal lifetime, it would sacrifice its nodal lifetime to accommodate the extra delay by reducing $T_r(p)$. The calculation of $L'(j)$ is the same as in Case 2, while $L'(i)$



(a) Case 2: before route switch. (b) Case 2: after route switch.

(c) Case 4: before route switch. (d) Case 4: after route switch.

(e) Case 5: before route switch. (f) Case 5: after route switch.

Fig. 4. Examples of inter-route coordination.

and $L'(p)$ may be estimated as follows:

$$\begin{cases} L'(i) = \frac{e(i)}{c(i)+f(i,j)\frac{T_r(p)+\Delta\mathcal{D}-T_r(j)}{2}P} \\ L'(p) = \frac{e(p)}{c(p)+\left(f(i,j)\left(2\tau+\frac{T_r(p\text{'s parent})}{2}\right)-\frac{\Delta\mathcal{D}\cdot\phi}{T_r(p)(T_r(p)+\Delta\mathcal{D})}\right)P} \end{cases} \tag{16}$$

An example is given in Figures 4(c) and 4(d), where $\Delta\mathcal{D} = -0.4s$ is accommodated by p through reducing $T_r(p)$ from $0.9s$ to $0.5s$. As a result, the minimal nodal lifetime between i, j, and p is actually decreased after the route switch. Therefore, i shall not change its parent node in this example.

Case 5: $L(i) \geqslant L(p) > L(j)$ and $\Delta\mathcal{D}_{\text{leaf}\to i\to p\to\text{sink}} < 0$. In this case, as i has a longer nodal lifetime, it will sacrifice its nodal lifetime to accommodate the extra delay by reducing $T_r(i)$. The calculations of the predicted nodal lifetimes are the same as in Case 2. An example is given in Figures 4(f) and 4(g). In this example, as i has a relatively long nodal lifetime, it successfully accommodates the extra delay incurred by the new route, and improves the minimal nodal lifetime between i, j, and p from 20h to 21h.

C. Design Discussion

1) Handling of Packet Losses: When the channel condition deteriorates, data or ACK packets may get lost, and the sensor node may need to retransmit multiple times before the data packet can be delivered successfully. As a result, the end-to-end delivery delay may exceed the delay bound. This issue can be dealt with by extending the I[2]C scheme by including ETX(i, j) – the expected number of transmission attempts to deliver a data packet successfully from i to j – in the design and analysis of the scheme. For example, the end-to-end delivery delay in Equation (4) would become

$$\mathcal{D}_{\text{src}\to\text{sink}} = \sum_{\texttt{all hops from source to sink}} T_r(j) \cdot \text{ETX}(i,j). \tag{17}$$

This way, a deteriorated channel condition with an increased ETX can be accommodated by reducing the corresponding T_r. Similarly, the lifetime estimation in Equations (1) and (2) can also be modified to include the ETX information. The value of ETX(i, j) can be estimated based on the periodical exchanges of beacons between neighbors for the routing purpose, as has been implemented in the CTP [23] protocol.

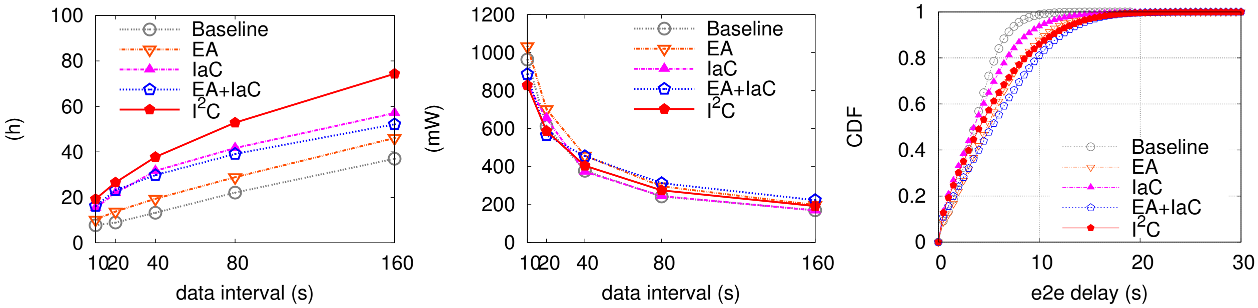
2) Handling of Routing Loops: The Inter-Route Coordination module of the I[2]C scheme handles the routing loops as follows. Firstly, when a node chooses a routing parent, any node that currently uses the node as its parent will not be considered. Secondly, when a node detects that the sum of delay from itself to the sink and delay from leaf to itself is larger than the end-to-end delay bound, while these reported delay values keep increasing but with a fixed T_r at its parent node, it considers that a routing loop has been detected; subsequently, the node's current parent node will be blacklisted for several rounds of data transmissions, and a new parent node is selected instead.

3) Handling of Child Leaving and Joining: After a child node has switched to a different parent node, its previous parent node may keep using the old T_r value that was selected to work with this child node. If this T_r value is small, the parent node wastes energy due to unnecessary short wake up intervals; if this value is large, it may take longer time for a newly joined child node to transmit data packets. In I[2]C, each node checks its children nodes periodically to evict stale ones from its children set. When a node becomes a leaf node, it will reset its T_r to the default value.

V. Performance Evaluation

NS-2 based simulations and TinyOS based testbed experiments have been conducted to evaluate the performance of the proposed I[2]C scheme terms of *network lifetime*, *network power consumption*, and *end-to-end delivery delay*. Here, network power consumption is defined as the total amount of energy consumed by the entire network of sensor nodes divided by the network lifetime. We compare the performance of I[2]C with the following representative combinations of energy-aware routing and intra-route coordination schemes.

- *CTP + RI-MAC (denoted as "Baseline" in figures):* The routing protocol is a customized CTP (Collection Tree Protocol) [23] which is modified to work in duty cycle networks and is able to satisfy the end-to-end delay requirement when selecting routing paths. The underlying MAC protocol is RI-MAC [20], and in the evaluation, T_r is 2 seconds and ϕ is 25 ms. This combination serves as the baseline scheme in the evaluation.
- *CTP + Intra-route Coordination (denoted as "IaC" in figures):* The routing protocol is the same modified CTP as in the baseline scheme. Intra-route coordination refers



(a) Network lifetime. (b) Network power consumption. (c) e2e delay.

Fig. 5. Performance comparison under different data generation intervals with uniform initial nodal energy distribution. The e2e delay requirement is 30 seconds and the total number of nodes in the network is 50.

to the Intra-Route Coordination module presented in Section IV-A where the MAC parameter T_r is adjusted to balance nodal lifetime between neighbor nodes. This combination evaluates the effectiveness of intra-route coordination only.

- *Energy-Aware Routing + RI-MAC (denoted as "EA" in figures):* In this combination, the energy-aware routing is adopted in the routing layer where each node selects the parent node that has the longest nodal lifetime from its neighbor set. In addition, only the routing paths that satisfy the end-to-end delay requirement may be selected. This combination evaluates the effectiveness of energy-aware routing only.
- *Energy-Aware Routing + Intra-route Coordination (denoted as "EA+IaC" in figures):* This is a simple combination of energy-aware routing and the Intra-Route Coordination module presented in Section IV-A. Different from our proposed holistic I^2C scheme, energy-aware routing and intra-route coordination simply co-exist in this combination without collaborating with or even being aware of each other.

A. NS-2 Simulations

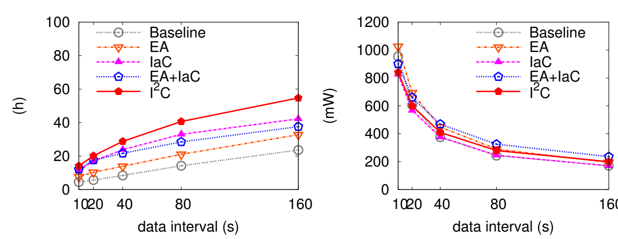
In the simulation, source nodes are randomly deployed in a 500m$\times$500m area and the sink is located at the center of the area. The evaluation results are averaged over results obtained in ten different random topologies.

We vary the data generation interval, the end-to-end delay requirement and the network density under different initial energy distributions. When the initial energy distribution is uniform, the initial nodal energy is full at 1000 Joules; when the distribution is non-uniform, the initial nodal energy is between 500 Joules and 1000 Joules at random. The maximal communication range is 70 meters and the power consumption is 69 mW when radio is on. In both simulations and testbed experiments, the default value of T_r is 2 seconds, the minimal value of T_r is 500 ms, and the routing update interval adopts the default setting in CTP.

1) Performance under Different Data Generation Intervals: Figures 5 and 6 compare the performances of all the evaluated schemes when the data generation interval at source nodes varies from 10 to 160 seconds.

As shown in Figure 5(a), I^2C always yields a longer network lifetime than other schemes. Particularly, when the data generation interval is 10 seconds (i.e., heavy workload scenario), I^2C extends the network lifetime by about 20% longer than the EA+IaC scheme, and 90% longer than the baseline scheme. When the data generation interval is 160 seconds (i.e., light workload scenario), the improvement on the network lifetime is about 40% over the EA+IaC scheme. The reasons behind the phenomena are explained as follows. The energy-aware routing allows nodes to choose routes of higher level of residual energy, but it may not be able to reduce workload for the bottleneck nodes on selected routes (for example, due to certain topology constraint) and therefore the network lifetime is bounded by these nodes. The intra-route coordination, on the other hand, can reduce the workload on the bottleneck nodes through shifting the workload to other nodes on the same route that have a longer nodal lifetime; however, it cannot coordinate the usage of nodes across routes, which constrains its capability in network lifetime prolonging. The above phenomena make it evident the necessity of integrating the two approaches.

A simple combination of the two approaches (i.e., EA+IaC), however, is shown to yield even a lower network lifetime than IaC under certain scenarios. This is because, without the awareness of intra-route coordination, the energy-aware routing protocol simply directs a sensor node to switch to a new parent node with a higher nodal lifetime. This may result in a lower network lifetime after intra-route coordination takes effect between the sensor node and its new parent node. Figure 4(c) and (d) in Section IV-B show an example of such scenarios, and explanation can be found in Section IV-B, Case 4. On the contrary, the intra-route coordination module of I^2C works with an inter-route coordination module that is well aware of intra-route coordination. As a result, I^2C inherits the advantages of both approaches and meanwhile mitigates



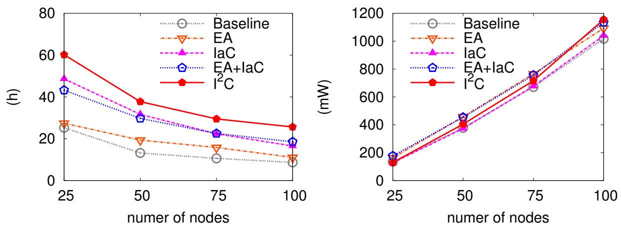
(a) Network lifetime. (b) Network power consumption.

Fig. 6. Performance comparison with different data generation intervals under non-uniform initial nodal energy distribution. The e2e delay requirement is 30 seconds and the total number of nodes in the network is 50.

their drawbacks, and therefore is shown to yield a significantly longer network lifetime than other schemes.

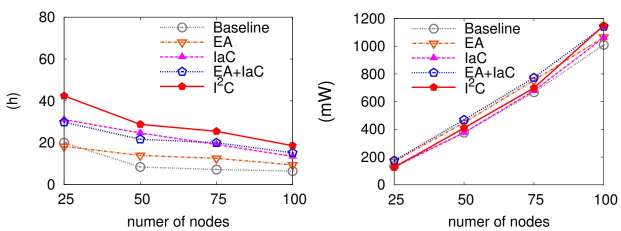
Figures 5(b) and 5(c) demonstrate that I^{2}C does not compromise its performance in other aspects, such as the end-to-end delay and the network power consumption. Due to space limitation, we omit the results of the end-to-end delay for other evaluation scenarios, where all the evaluated schemes satisfy the delay requirement – similar to what has been shown in Figure 5(c). Moreover, Figure 6 show that I^{2}C also performs consistently better than other schemes under the non-uniform initial nodal energy distribution as well.

2) Performance under Different Network Densities: The performance when the network density varies is demonstrated in Figures 7 and 8. As we can see from these figures, when the network density varies (i.e., the number of nodes in the network changes from 25 to 100), I^{2}C always yields a significantly longer network lifetime than other schemes while maintaining a similar level of network power consumption.



(a) Network lifetime. (b) Network power consumption.

Fig. 7. Performance comparison with different network densities under uniform initial nodal energy distribution. The e2e delay requirement is 30 seconds and the data generation interval is 40 seconds.

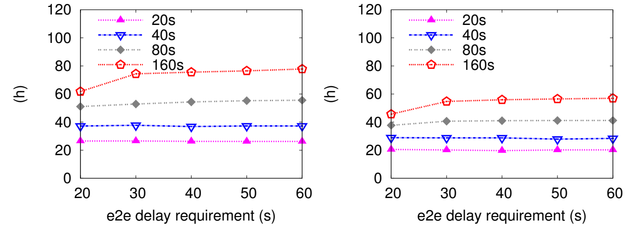


(a) Network lifetime. (b) Network power consumption.

Fig. 8. Performance comparison with different network densities under non-uniform initial nodal energy distribution. The e2e delay requirement is 30 seconds and the data generation interval is 40 seconds.

3) Performance under Different e2e Delay Requirements: We also evaluate the performance of I^{2}C when both the data generation interval and the end-to-end delay requirement vary.

From Figure 9 we can see that, when the data generation interval is short (i.e., 20 seconds), the achieved network lifetime does not change much as the delay requirement increases. This is because, when the network workload is heavy, the energy consumption on data transmissions, rather than the cost on periodic wakeup for data receptions, dominates the nodal energy consumption. In this case, a node can only increase its wakeup interval T_r to a certain value, as too large a T_r value may cause considerably more energy consumption for its children nodes according to the analysis in Equations (1) and (2) in Section III-B. Consequently, even with a relaxed end-to-end delay requirement, the change of T_r remains small;



(a) Network lifetime under uniform initial energy distribution. (b) Network lifetime under non-uniform initial energy distribution.

Fig. 9. Performance comparison with different e2e delay requirements. The total number of nodes in the network is 50. Different curves correspond to different data generation intervals.

that is, the opportunity for nodal lifetime balancing brought by the relaxation of delay requirement may not be fully utilized.

On the other hand, when the data generation interval is long (i.e., 160 seconds), the attained network lifetime increases when the end-to-end delay requirement is relaxed. This is because, when the network workload is light, the periodic wakeup and channel checking activities (i.e., $\frac{\phi}{T_r}$ in Equation (2)) becomes the dominant factor in nodal energy consumption. Therefore, a node can adjust its T_r in a larger range without causing much overhead on its children nodes' energy cost for data transmissions. This way, the lifetime balancing between parent and children nodes can be conducted more efficiently.

To summarize, ns-2 simulation results clearly demonstrate the consistent performance improvement of I^{2}C over the state-of-the-art solutions on prolonging the network lifetime under various network conditions.

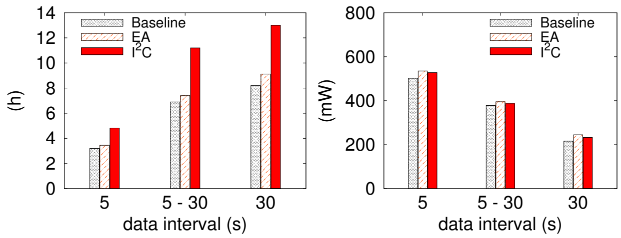
B. Testbed Experiments

1) Implementation: We have implemented I^{2}C in TinyOS 2.1.0. In our implementation, we modify the following sensor network messages to embed the needed control information. (i) Each *data message* carries a node's lifetime and the longest delivery delay from its leaf nodes to the node itself. (ii) Each *ACK message* carries a node's T_r value and the delivery delay from the node to the sink. (iii) Each *periodic routing update message* carries a node's residual energy and energy consumption rate, the T_r values of the node itself and its parent node, as well as the delivery delay from the node to the sink.

2) Testbed Setup and Evaluation Results: We set up a testbed network of 37 TelosB motes to evaluate the performance of the proposed scheme. In the testbed network, 36 nodes are placed in a 6×6 grid topology where the distance between two adjacent nodes is about 2 meters. All these nodes are source nodes and produce sensory data periodically. An extra node is placed near the upper left corner of the grid; it is connected to a PC and keeps its radio on all the time to serve as the sink. In the experiments, we compare the performance of I^{2}C with the the Baseline and EA schemes. The end-to-end delivery delay requirement is 30 seconds.

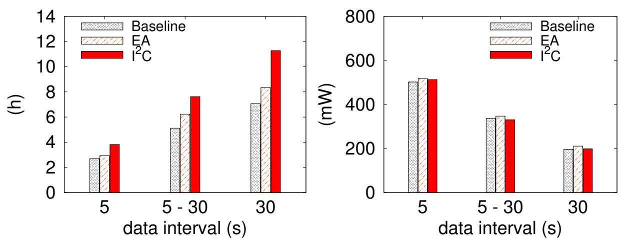
In order to complete the experiments within a reasonable amount of time, we study how fast a node consumes a small designated amount of energy, and evaluate its nodal lifetime as the time period during which the designated amount of energy is consumed. The network lifetime is the minimal nodal lifetime among all sensor nodes. At the beginning of each experiment, the initial nodal energy distribution is uniform or non-uniform. When the distribution is uniform, the initial available energy at an individual node is designated to 400 Joules; when it is non-uniform, the initial available energy at an individual node is designated to a random value between 250 Joules and 400 Joules.

As can be seen from Figures 10 and 11, in the testbed network, the performance improvement achieved by the EA scheme over the Baseline scheme is limited due to the bottleneck effect. However, I^{2}C still yields a significant longer network lifetime than both EA and Baseline schemes under different network traffic loads and initial energy distributions.



(a) Network lifetime. (b) Network power consumption.

Fig. 10. Experiment results with different data generation intervals under uniform initial nodal energy distribution. Data interval "5-30" means that data packets are generated at an interval uniformly distributed in [5s, 30s].



(a) Network lifetime. (b) Network power consumption.

Fig. 11. Experiment results with different data generation intervals under non-uniform initial nodal energy distribution. Data interval "5-30" means that data packets are generated at an interval uniformly distributed in [5s, 30s].

VI. Conclusions and Future Work

In this paper, we present I^{2}C – a new holistic approach to prolong the sensor network lifetime. I^{2}C is composed of two collaborative modules: intra-route coordination and inter-route coordination modules. Different from most of the existing works which conduct either intra-route or inter-route lifetime balancing alone, I^{2}C leverages and integrates the advantages of both approaches and therefore can prolong the network lifetime more efficiently. In addition, I^{2}C can also meet the end-to-end delay requirement specified by the applications. Extensive simulation and testbed experiments have been conducted, and the evaluation results show that I^{2}C can significantly prolong the network lifetime than the state-of-the-art solutions.

In the future, we will improve the I^{2}C design by tuning more MAC parameters such as channel checking and data retry intervals. We are also going to explore the feasibility and strategies of embracing data aggregation [24] into the I^{2}C design, such that the network lifetime may be prolonged further. Besides, we also plan to extend the I^{2}C design to networks with different traffic patterns such as broadcast or multicast.

Acknowledgement

This work was supported partly by the NSF under grant CNS- 0831874 and ECCS-1128312, and by the ONR under grant N00014-09-1-0748.

References

[1] J.-h. Chang and R. Tassiulas, "Energy Conserving Routing in Wireless Ad-hoc Networks," in *IEEE INFOCOM '00*, 2000, pp. 22–31.

[2] G. W. Challen, J. Waterman, and M. Welsh, "IDEA: Integrated Distributed Energy Awareness for Wireless Sensor Networks," in *ACM MobiSys '10*, 2010, pp. 35–48.

[3] J. Park and S. Sahni, "An Online Heuristic for Maximum Lifetime Routing in Wireless Sensor Networks," *IEEE Trans. Comput.*, vol. 55, pp. 1048–1056, August 2006.

[4] Q. Li, J. Aslam, and D. Rus, "Online Power-aware Routing in Wireless Ad-hoc Networks," in *ACM MobiCom '01*, 2001, pp. 97–107.

[5] L. Lin, N. B. Shroff, and R. Srikant, "Asymptotically Optimal Power-Aware Routing for. Multihop Wireless Networks with Renewable Energy Sources," *IEEE/ACM Trans. Netw.*, vol. 15, no. 5, pp. 1021–1034, October 2007.

[6] R. Braynard, A. Silberstein, and C. S. Ellis, "Extending Network Lifetime Using an Automatically Tuned Energy-Aware MAC Protocol," in *Springer-Verlag EWSN '06*, 2006, pp. 244–259.

[7] Z. Li, M. Li, and Y. Liu, "Towards Energy-Fairness in Asynchronous Duty-Cycling Sensor Networks," in *IEEE INFOCOM '12*, 2012, pp. 801–809.

[8] Y. Peng, Z. Li, W. Zhang, and D. Qiao, "LB-MAC: A Lifetime-Balanced MAC Protocol for Sensor Networks," in *International Conference on Wireless Algorithms, Systems, and Applications (WASA '12)*, 2012.

[9] Y. Gu and T. He, "Data Forwarding in Extremely Low Duty-Cycle Sensor Networks with Unreliable Communication Links," in *ACM SenSys '07*, 2007, pp. 321–334.

[10] ——, "Dynamic Switching-Based Data Forwarding for Low-Duty-Cycle Wireless Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 10, pp. 1741–1754, 2011.

[11] A. Meier, M. Woehrle, M. Zimmerling, and L. Thiele, "ZeroCal: Automatic MAC Protocol Calibration," in *IEEE DCOSS '10*, 2010, pp. 31–44.

[12] M. Zimmerling, F. Ferrari, L. Mottola, T. Voigt, and L. Thiel, "pTunes: Runtime Parameter Adaptation for Low-power MAC Protocols," in *ACM IPSN '12*, 2012, pp. 173–184.

[13] M. Conti, G. Maselli, G. Turi, and S. Giordano, "Cross-Layering in Mobile Ad Hoc Network Design," *IEEE Computer*, vol. 37, pp. 48–51, 2004.

[14] V. Kawadia and P. R. Kumar, "A Cautionary Perspective on Cross Layer Design," *IEEE Wireless Communications*, vol. 12, pp. 3–11, 2005.

[15] S. Cui, R. Madan, A. Goldsmith, and S. Lall, "Joint Routing, MAC, and Link Layer Optimization in Sensor Networks with Energy Constraints," in *IEEE ICC '05*, 2005, pp. 725–729.

[16] R. Madan, S. Cui, S. Lall, and N. Goldsmith, "Cross-Layer Design for Lifetime Maximization in Interference-Limited Wireless Sensor Networks," *IEEE Transactions on Wireless Communications*, vol. 5, no. 10, pp. 3142–3152, 2006.

[17] H. Wang, X. Zhang, F. Nait-Abdesselam, and A. Khokhar, "Cross-Layer Optimized MAC to Support Multihop QoS Routing for Wireless Sensor Networks," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 5, pp. 2556–2563, 2010.

[18] S. Ehsan, B. Hamdaoui, and M. Guizani, "Cross-Layer Aware Routing Approaches for Lifetime Maximization in Rate-Constrained Wireless Sensor Networks," in *IEEE GLOBECOM '10*, 2010, pp. 1–5.

[19] L. Chen, S. H. Low, M. Chiang, and J. C. Doyle, "Cross-Layer Congestion Control, Routing and Scheduling Design in Ad Hoc Wireless Networks," in *IEEE INFOCOM '06*, 2006, pp. 1–13.

[20] 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 *ACM SenSys '08*, 2008, pp. 1–14.

[21] Y. Gu, T. He, M. Lin, and J. Xu, "Spatiotemporal Delay Control for Low-Duty-Cycle Sensor Networks," in *IEEE RTSS '09*, 2009.

[22] S. Guo, Y. Gu, B. Jiang, and T. He, "Opportunistic Flooding in Low-Duty-Cycle Wireless Sensor Networks with Unreliable Links," in *ACM MobiCom '09*, 2009, pp. 133–144.

[23] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection Tree Protocol," in *ACM SenSys '09*, 2009, pp. 1–14.

[24] Z. Li, Y. Peng, D. Qiao, and W. Zhang, "LBA: Lifetime Balanced Data Aggregation in Low Duty Cycle Sensor Networks," in *IEEE INFOCOM '12*, 2012, pp. 1844–1852.