OSC-MAC: Duty Cycle Scheduling and Cooperation in Multi-hop Wireless Sensor Networks

Jian Lin and Mary Ann Ingram
School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, Georgia 30332–0250
Email: jlin61@gatech.edu; mai@gatech.edu

Abstract—**An On-demand Scheduling Cooperative MAC protocol (OSC-MAC) is proposed to address the *energy hole* problem in multi-hop wireless sensor networks (WSNs). By combining an on-demand strategy and sensor cooperation intended to extend range, OSC-MAC tackles the spatio-temporal challenges for performing cooperative transmission (CT) in multi-hop WSN: cooperating nodes are neither on the same duty cycle nor are they in the same collision domain. We use orthogonal and pipelined duty-cycle scheduling, in part to reduce traffic contention, and devise a reservation-based wake-up scheme to bring cooperating nodes into temporary synchrony to support CT range extension. Compared with existing MAC protocols, our NS-2 simulations with REACT show that while we explicitly account for the overhead of CT and practical failures of control packets in dense traffic, OSC-MAC still gives almost 80% increase in network lifetime.**

I. INTRODUCTION

The *energy hole* is known to limit the lifetime of battery-powered multi-hop wireless sensor networks (WSNs) [1]. The hole occurs when nodes near the sink exhaust their energy first because their load is heavier: they must transmit packets they originate and relay packets from and to other nodes farther away from the sink. The Residual Energy Activated Cooperative Transmission (REACT) [2] forwarding scheme has been shown through simulation to correct the energy imbalance and extend network lifetime by factors of 2 or more by exploiting cooperative transmission (CT) range extension. However, the increased control overhead (OH) from CT was not considered in [2]. Also, [2] did not consider how REACT could be merged with duty cycling, which is a popular way to conserve energy and extend the life of a WSN. This paper proposes a novel and practical Medium Access Control (MAC) protocol that supports both REACT and duty cycling, and properly accounts for the control packet OH. Even with the OH, NS-2 simulation results show the MAC protocol with REACT produces an 80% increase in network lifetime.

As indicated by [3], the major sources of energy consumption inherent to MACs include idle listening, overhearing and collision, besides data transmission and reception. Idle listening means that nodes keep listening to the channel while there are no incoming packets at all - a case that has not been taken care of in many MAC protocols such as IEEE 802.11 [4] wherein WIFI stations must listen for possible traffic. Notably, idle listening is disastrous in WSNs based on the fact that nodes in this mode consume the same magnitude of power as in receiving [3]. Overhearing means that nodes decode packets that are destined to others. Collisions result in corrupted packets and the following MAC layer retransmissions consume extra energy. From the network perspective, albeit these factors taper off individual node's lifetime, the network lifetime is more critically limited by the *energy holes* formed around the sink leaving unused energy outside of the holes.

Many authors have considered duty-cycle MAC protocols [3] [5] [6] [7], which allow nodes to alternate between active and sleep modes. These protocols dramatically reduce the periods of idle listening and overhearing. In particular, DW-MAC [5] has been shown to have superior delivery ratio, delay, and energy consumption. However, DW-MAC and the others do not solve the energy hole problem in multi-hop WSNs.

Meanwhile, many other authors have proposed MACs to support CT [8] [9] [10] [11]. CT takes place when multiple single-antenna radios coordinate to form a virtual antenna array, thereby enabling a dramatic signal-to-noise ratio (SNR) advantage at the receiver. Yet none of these works addresses network lifetime or alleviates the *energy hole* problem. Further, they all make the spatial assumption that the source, cooperators and the destination are located in one collision domain (i.e. within transmission range of each other), and the temporal assumption that all nodes stay active when CT is performed. The spatial assumption is not generally true for multi-hop networks, and the temporal assumption is not generally satisfied in asynchronous duty-cycle networks.

The CT SNR advantage can be used to extend range. The REACT protocol [2] exploits this to "hop-over" the nodes near the sink, thereby avoiding the energy hole. However, the analysis of REACT did not consider the impact of the control packets, in part because there did not exist a MAC protocol to support REACT. Motivated by REACT, SCT-MAC [12] and CDC-MAC [13] were our efforts to design a MAC for REACT and to bring CT into the duty cycle context. In SCT-MAC, to protect the one-hop parent, a node transmits directly to its two-hop parent by incorporating one cooperator; also the control packet from the two-hop parent has to traverse two non-CT hops back to the child. A disadvantage of SCT-MAC is that it requires some nodes to maintain up to three schedules, producing more than necessary wakeup periods to support CT. CDC-MAC [13] also considers CT in a duty cycle

The authors gratefully acknowledge support for this research from the National Science Foundation under grant CNS-1017984.

978-1-4673-5939-9/13/$31.00

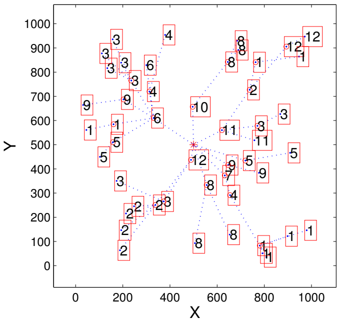

Figure 1: A scheduling instance of random topology

context, however it addresses only a two-hop network and lacks scalability to the multi-hop network.

Different with most papers on MAC that focus on either an *energy-conserving* or an *energy-balancing* strategy, we propose an On-demand Scheduling Cooperative MAC (OSC-MAC) to incorporate both heuristics in MAC design, to achieve a more balanced energy consumption and to extend the network lifetime. In particular, we solve the energy hole problem through a MAC that seamlessly schedules CT range extension, and relaxes the temporal assumption via an on-demand wake-up scheme. To support the above, a pipelined and orthogonal schedule embedded in the MAC is proposed to enable on-demand CT and to reduce traffic contention.

The rest of this paper is structured as follows. Section II describes the network model and the duty cycle scheduling. Section III presents different aspects of the proposed OSC-MAC. Section IV evaluates the performance of our protocol compared with DW-MAC and SCT-MAC by NS-2 simulations. The concluding remark is given in Section V.

II. NETWORK MODEL AND DUTY CYCLE DESIGN

As required by REACT, a tree-based routing protocol is assumed. We consider Time Division CT (TDCT) that includes a multi-cast phase and several uni-cast phases [8]. In the first multi-cast phase, the source shares one copy of the packet to the cooperators. Subsequently, the cooperators forward the the packet to the destination in orthogonal time slots. In order to achieve CT range extension, the radio of the source node, cooperators and the destination must be on at the time when the source firstly multi-casts the packet to the cooperators[1]. Our previous work SCT-MAC provides the wakeup rendezvous by letting a node wake up during its parent's and two-hop parent's schedule, namely $Sched\,1$ and $Sched\,2$, and initiating CT during $Sched\,2$. However, in this scheme the nodes far from the sink usually have to maintain more schedules than the nodes near the sink, and therefore consume more energy in idle listening. In addition, the candidate helpers are limited to the siblings who share a common parent or the parent's siblings, because of the schedule. In this study, we present a different approach to bring the cooperating nodes into temporary synchrony in an on-demand fashion. Specifically, each node shall maintain a permanent schedule of its own to receive incoming packets, and wake up only on demand to support CT, which we call temporary schedules.

One *duty-cycle* is composed of the scheduling period, the data period, and the sleep period. The *superframe* is defined as the concatenation of the scheduling period and the data period. We define the length of a *cycle* to be N_s superframes. During a superframe, the scheduling period is used for cooperation wakeup request and transmission reservation. The non-CT data transmissions and CT data transmissions are performed in the data period according to the scheduling information obtained in the preceding scheduling period. Nodes sleep during the entire sleep period, and also sleep in the "unused" portion of the data period. We use the same greedy algorithm and broadcast procedure at network initialization phase as SCT-MAC, to establish the permanent superframe schedules as in Alg.1. The resulting schedules in a path from a low-level node towards the sink are sorted in a cyclic increasing order. The pipelined feature is similar to [7]. Unlike [7], the schedules of interference nodes appear orthogonal in time (i.e., the superframe of a node lies in the sleep period of its interfering nodes). Here we assume the inference range is twice of SISO transmission range, which has been validated by measurements [14]. The introduced orthogonality guarantees that different traffic flows in the network are collision free. Same as DW-MAC [5], a separate network time synchronization protocol is assumed to disseminate the duty cycle information. Fig. 1 shows an instance of the scheduling algorithm for a random topology of 50 nodes with $N_s = 12$. The number in the square represents the permanent schedule of the corresponding node. Note that a leaf node follows its parent's schedule.

Conducting CT in an asynchronous network is extremely challenging, because the source, the cooperators and the destination need to reach consensus about a wake-up period, during which CT can be performed. In this paper, we explore an explicit requesting procedure to gather the cooperators and the one-hop parent to a common wakeup rendezvous, which will be described in the next section.

Algorithm 1: Superframe Scheduling

```
input:
$p_t$ = Direct parent of Node $t$
$I_t$ = Interfering parent set of Node $t$
$S_a = \{s \in [1, Ns] : s \neq s_i \; \forall\, i \in I_t,\; s_i$ is the schedule of Node $i\}$
output: $s_t$
begin
    $S_a^d \leftarrow$ sort $S_a$ in descending order
    for $s \in S_a^d$ do
        if $s < s_{p_t}$ then $s_t \leftarrow s$,  break ;
        if $s = \min S_a^d$ then
            $s_t \leftarrow \max S_a^d$;
        end
    end
    return $s_t$
end
```

---

[1] The destination needs to be on so that it can sample and store the signal received from the source, to combine later with copies received from the cooperators.

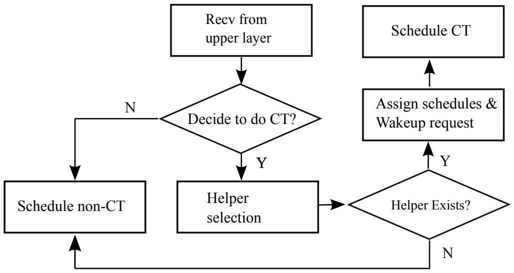

Figure 2: CT decision and wake-up scheduling at the source

III. ON-DEMAND SCHEDULING COOPERATIVE MAC

A. Overview

In OSC-MAC, each node maintains one permanent schedule and decides temporary schedules on the fly. The temporary schedules are activated on demand to support CT and deactivated afterwards. It is assumed that each node is aware of the permanent schedules of its one-hop neighbors and its two-hop parent. Because every node can be a receiver, it always keeps awake during its own permanent schedule to receive either CT data or non-CT data. When a node decides to transmit a non-CT data packet directly to its parent, it wakes up in its parent's permanent schedule and proceeds to transmit using the procedure in Subsection D. On the other hand, when the source node decides to do CT, as shown in Fig. 2, it tries to filter the candidate cooperators applying the criteria in Subsection B. Here we use the same simple criterion for CT decision as in REACT, which compares the source node's residual energy E_i and its parent's energy E_p. If $E_p > E_i$, a non-CT transmission will be performed. Otherwise, a CT decision will be made to protect the parent. Then the cooperating nodes will decide temporary schedules on the fly, during which they will wake up. An algorithm is provided in Subsection C to achieve this. To support the scheduling in OSC-MAC, three classes of active schedule are defined, and the responsibility of the corresponding nodes are listed as follows:

- *Permanent schedule (Class 1)*: Set by each node to listen for incoming packets.
- *Temp schedule (Class 2)*: Set by source node that initializes CT, to perform wakeup request to cooperators.
- *Temp schedule (Class 3)*: Set by cooperating nodes as wakeup rendezvous, to perform CT.

B. Cooperator Selection

In contrast to SCT-MAC, which requires that the cooperator must be the sibling of the source node or the sibling of the parent due to the scheduling design, OSC-MAC does not have such constraint. Instead, each node can choose the cooperator from its one-hop neighborhood. In this study, we consider selecting the helper that has: 1) the maximum residual energy, and 2) energy that is higher than the source node. Energy information can be obtained by inserting a common information-sharing broadcast slot periodically. Packets transmitted during the broadcast period can also be used to readjust permanent schedules and update neighbor information, to accommodate possible topology variations.

C. On-demand Wakeup

In the case of CT, we present the on-demand wakeup scheme to bring cooperating nodes into temporary synchrony, by managing temporary schedules on the fly. Specifically, we require that CT should be performed during the two-hop parent's permanent schedule, which is the schedule had a two-hop non-CT been performed. Thus, the objective is to have all the cooperating nodes locked onto this CT rendezvous.

Because nodes may have distinct permanent schedules, an explicit wakeup request procedure is proposed to set up the wakeup rendezvous for the cooperating nodes. As in Fig. 3, the source node i sets $Class$ 2 schedules to temporarily wake up in each of the cooperator's and the one-hop parent's schedule s_j. Note that each s_j is a concatenation of a scheduling period and a data period. The source node will send a wakeup request packet to the cooperating nodes to indicate the expected time of CT rendezvous β, and wait for the wakeup replies. Depending on Slot α when the packet is generated and the schedule of the cooperating nodes, a method is provided for the cooperating nodes to self-decide when to wake up to perform CT, as shown in Alg.2. This coordination process falls in the *non-CT* case of the MAC procedure as will be described in Subsection D, i.e., with the scheduling period used for exchanging wakeup request/reply.

Algorithm 2: On-demand Wakeup Schedules

```
Input: Set of cooperating nodes (including the parent) $\Gamma$
       The slot when packet arrives, $\alpha$
       The schedule of the two-hop parent, $\beta$
Output: The schedule of wakeup rendezvous
Source: (do wakeup request)
$maxTime \leftarrow 0$ ;
foreach $j \in \Gamma$ do
    if $s_j \equiv \beta$ then $setWakedStatus[j]$ ;
    else
        $d_1 \leftarrow (s_j - \alpha) \bmod Ns$ ;
        $d_2 \leftarrow (\beta - s_j) \bmod Ns$ ;
        $maxTime \leftarrow \max\{maxTime, startofSlot + (d_1 + d_2) * superFrameLen\}$ ;
        $scheduleTxRequest[j]$ ;
    end
end
Each cooperating node: (when receiving wakeup request)
$d \leftarrow (\beta - s_j) \bmod Ns$ ;
$TimeToWake \leftarrow startofSlot + d * superFrameLen$ ;
if $TimeToWake < maxTime$ then $scheduleDelayedWakeup()$;
else
    $scheduleWakeup()$ ;
end
$sendWakeupReply()$ ;
```

Remark: There are two kinds of delay: (a) link delay (from the source to the next-hop, either one-hop parent or two-hop parent); (b) possible "extra" delay resulting from cooperator coordination. Note that there will be no extra delay in the cooperator coordination process if the selected helper's on time happens before the two-hop parent's (referred as *coordination in advance*), because of the pipelined feature of permanent

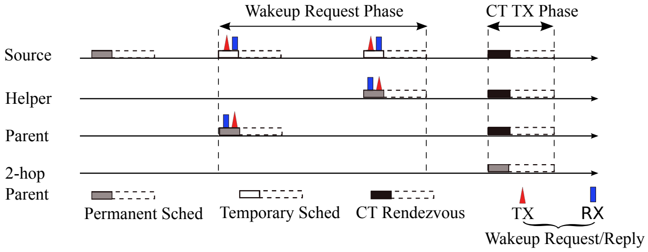

Figure 3: Wakeup scheme

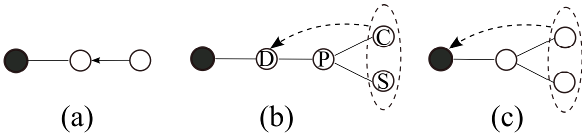

Figure 4: First-hop node traffic

schedules. Indeed, a third criterion can be incorporated into cooperator selection (i.e., $s_j < \beta$), to ensure *coordination in advance*. Our results show that OSC-MAC has similar link delay as SCT-MAC, which is about twice of DW-MAC, exhibiting the inevitable tradeoff between CT and delay. Thus, our protocol favors applications that can tolerate certain delays [15], but would benefit from a significantly longer lifetime, such as ecology monitoring (e.g. lake water quality, forest temperature monitoring), structural health monitoring or animal habit tracking.

D. Seamless Scheduling and Transmission for CT and Non-CT

In this subsection, we provide the details for the seamless scheduling and transmission for non-CT during the parent's schedule, and for CT during the CT rendezvous as obtained in Fig. 3 (CT TX Phase). Same as SCT-MAC, the scheduling period is used to reserve the data transmission in the following data period. Nodes fall asleep during the unused portion of the data period. A receiver will keep track of the numbers of non-CT and CT transmissions that have already been reserved in its own data period.

1) Non-CT case: During the scheduling period, the source node S transmits a scheduling frame (SF) to the destination D, which will reply with a SF including the numbers of non-CTs (N_{non-CT}) and CTs (N_{CT}) that have already been reserved in the data period. Both Node S and Node D determine the corresponding wakeup time instance T_{Wakeup} from the beginning of the subsequent data period, according to:

$$T_{Wakeup} = T_{non-CT} \times N_{non-CT} + T_{CT} \times N_{CT} \quad (1)$$

where, $T_{non-CT} = T_{Pkt} + T_{Ack} + SIFS$ and $T_{CT} = 2T_{Pkt} + 2T_{Ack} + 3SIFS$. It is easy to see that for a particular receiver, no schedule conflicts will occur during the subsequent data period, and thus data collision from different transmitters is avoided. After waking up at the scheduled time T_{Wakeup} in the data period, Node S and Node D carry out a DATA-ACK process, and then fall asleep again.

2) CT case: Same as SCT-MAC, one cooperator is used to achieve doubling range extension. Using the topology in Fig. 4 (b) as an example, during the period of CT rendezvous, Node S sends a SF to Node D which is a two-hop away destination. The enlisted cooperator C then forwards the SF to Node D in the next time slot after SIFS. Node D, after decoding the SFs using Maximum Ratio Combining[2] (MRC) [16], replies to the source S with a SF including N_{non-CT} and N_{CT}. Since no CT is used in the destination's transmission, this SF must be forwarded by the one-hop parent of the source node (P, which is to be *protected*). After receiving the forwarded SF, the source S and cooperator C will schedule a wakeup time according to Eq.1. Then in the wakeup time in the data period, the data packet is sent by S during the first time slot, which is decoded and forwarded by Node C during the second time slot after SIFS. The two-hop destination D when decoding the CT packets using MRC transmits ACK to Node S via two hops. Again, the ACK is forwarded by Node P on behalf of Node D. To avoid overhearing the DATA packets from Node S and Node C, Node P should schedule a wakeup time T_p before Node D transmits ACK, i.e., $T_p = T_{Wakeup} + 2T_{Pkt} + 2SIFS$. Therefore we cannot completely avoid the usage of P, which is to be protected; we use it only for forwarding the control packets and allow it to sleep otherwise.

Table I: Simulation Parameters

Bandwidth	20 Kbps	Channel Enc. Ratio	2
Tx Power	31.2 mW	Tx Range	250 m
Rx Power	22.2 mW	CS Range	550 m
Idle Power	22.2 mW	Superframe Length	3071 ms
Sleep Power	3 uW	Cycle Length	36.85 s
State Transition Power	31.2 mW	Size of ACK	10 B
DIFS	2 ms	Size of SF/CSF	14 B
SIFS	0.6 ms	Size of Data	100 B
Contention Window	16 ms	Retry limit	5

Retransmissions for CT and non-CT include the retries of SF handshake and DATA transmission. The former case occurs in the scheduling period which can hold several retries until reaching the boundary of the period. However, if the SF handshake succeeds but the scheduled data transmission fails in the data period, the node has to wait until the next cycle.

E. First-hop Node Handling and Fast Forwarding to the Sink

Since nodes that are one-hop from the sink have orthogonal schedules (because they are within interference range of each other), they see the sink in split time which makes it possible for them to send replying SF directly to the source, instead of letting the sink transmit to the source via two hops. There are three kinds of traffic going through a first-hop node as shown in Fig. 4 : the non-CT and CT transmission with itself as MAC layer destination as in (a) and (b), and the CT transmission from its leaf nodes directly to the sink as in (c). In the first two cases, the node reserves one more T_{non-CT}, which is used to forward the received data immediately to the sink. In the third case, it replies with SF on behalf of the sink.

---

[2] Channel estimates with respect to the collaborating nodes can be obtained from PLCP preambles.

IV. SIMULATION RESULTS

In this section, we present detailed performance results for the proposed OSC-MAC protocol from simulations using NS-2.29. We consider 100 random topologies where 50 nodes are randomly distributed in an area of 1000 m * 1000 m. The sink is located in the center. Same as SCT-MAC and DW-MAC, OSC-MAC is analyzed using the Random Correlated Event (RCE) traffic model to simulate burst traffic triggered by spatially correlated events. The event location is randomly selected every 200 seconds. Each node within the circle of radius R centered at the event location generates a packet to the sink. The radius R is gradually increased to input more traffic into the network. The main simulation parameters are listed in Table I. The initial energy of node is set to $50J$ and the sink has no energy constraint. The routing layer uses the minimum-distance metric and the energy consumption occurring in the routing layer is neglected. We also neglect the energy consumed during the network initialization phase to build the permanent schedules in OSC-MAC. The performance of OSC-MAC is compared with the cooperative protocol SCT-MAC [12] and the non-cooperative duty-cycle protocol DW-MAC [5].

A. Network Lifetime Evaluation

In this subsection, the evaluation of network lifetime of different protocols is given. The lifetime is defined as the number of packets that have been successfully delivered to the sink when the first node dies. Fig. 5 shows the growth trend of the average lifetime as the event sensing range R increases, with 95% confidence intervals (they are very small). We observe that OSC-MAC outperforms SCT-MAC and DW-MAC significantly in lifetime. For example, OSC-MAC increases the mean network lifetime by 77.8% compared with SCT-MAC when $R = 400m$. This is because OSC-MAC, in spite of CT, spends less time in idle listening. And when traffic increases, the staggered duty-cycles reduce the collisions when many nodes contend for the medium.

Fig. 6 shows in very small scale the average delivery ratio of the three protocols, with 95% confidence intervals. Besides having the highest delivery ratio, OSC-MAC delivers the smallest confidence intervals for all the event sensing ranges.

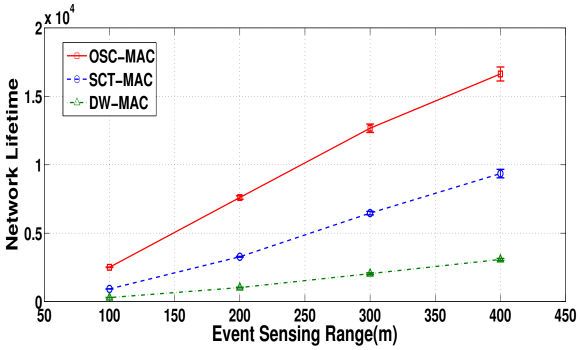

Figure 5: Network lifetime v.s. Event sensing range

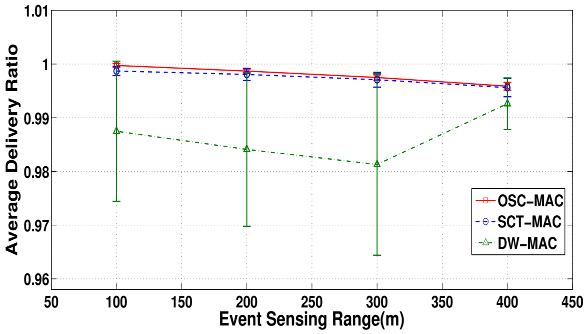

Figure 6: Delivery ratio v.s. Event sensing range

B. Residual Energy Profile

Here we show the residual energy profile of all the nodes in the network when the first node dies, for event sensing ranges of $100m$ and $300m$. Fig. 7 suggests that SCT-MAC leaves more energy around the sink unused. This is because the scheduling in SCT-MAC requires a node to wakeup in the parent's and two-hop parent's schedule to support CT, and thus nodes farther away from the sink maintain up to three wakeup schedules consuming more energy than necessary in idle listening. From Fig. 7, we also see that DW-MAC obtains a balanced residual energy profile, however, lots of the energies are consumed in idle listening in the scheduling period and channel contention because of the synchronized schedules of all the nodes. OSC-MAC, on the other hand, achieves more balanced energy than SCT-MAC because nodes are required to maintain one permanent schedule and manage temporary schedules on the fly, and therefore the periods spent in idle listening for possible CT traffic are reduced. However, the energy in OSC-MAC is less balanced than DW-MAC at the nodes near the sink. Also OSC-MAC still leaves a significant amount of average residual energy at first node death, suggesting more efficient protocols may be possible. The imbalance is due to the limitation of the two-hop range extension in our scheme, and also the practical failures of CT handshake as will be discussed in Subsection D. With longer range extension, the energy could be more balanced as shown in REACT [2]. However, longer range extension imposes more challenges in control packets exchange.

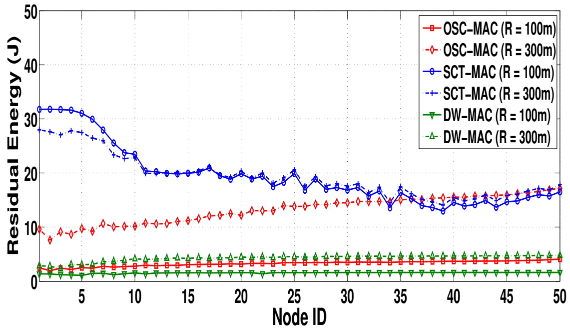

Figure 7: Residual energy profile when the first node has depleted its energy

C. Saturation Lifetime

To quantify the influence of CT in OSC-MAC protocol, we increase the sensing range from $100m$ to $1200m$ and compare the lifetime (with 95% confidence intervals) obtained in OSC-MAC with CT enabled and CT disabled cases. "CT disabled" means that the duty cycle assignment as in Section II is held, however every packet is forced to follow non-CT transmission. With sensing range of $1200m$ almost every node would transmit periodically, and hence there are 50 flows converging to the sink making the network heavily loaded. We observe in Fig. 8 that after an increasing trend before the sensing range reaches $600m$, the lifetime gradually decreases until arriving at a plateau. A similar trend of throughput as traffic increases is also observed by Bianchi [17] in the WIFI network. Similarly, we define the plateau as the saturation network lifetime. Fig. 8 shows that CT increases the saturation lifetime comparing with non-CT transmission even when both cases are under carefully designed duty cycle schedules and when the communication overhead of CT is considered.

D. Practical Limitation Due to CT Control Handshake

We observe that CT is not always being performed as desired. Fig. 9 plots the CT cancellation frequency, for every source node with event sensing ranges of $200m$ and $600m$. Note that the first-hop nodes to the sink have no need to initialize CT. We observe that when the traffic load in the network is heavy, CT cancellations occur much more frequently. This takes place for two reasons: (1) the SF handshake during CT rendezvous for scheduling CT data transmission suffers more collisions when the network is heavily loaded. The collisions come from the contention with both non-CT an CT handshakes; (2) as we implement an explicit wakeup request/reply procedure to reach CT rendezvous, this wakeup procedure could fail more frequently due to contentions. Consequently, the CT attempt must be canceled and subsequently the non-CT is pursued.

V. CONCLUSIONS

An on-demand duty cycling MAC (OSC-MAC) has been proposed to support the REACT protocol, which exploits CT range extension at the network layer to solve the *energy hole* problem in multi-hop WSNs. By combining an on-demand schedule and CT range extension, we have addressed the spatio-temporal challenges for performing CT in multi-hop WSNs, to offer significantly longer lifetime. Even with control packet energy accounted for, OSC-MAC with REACT still produces almost 80% increase in network lifetime.

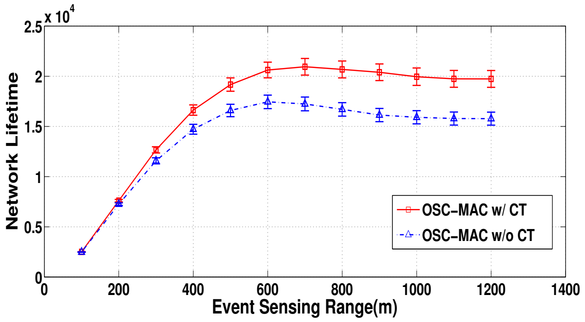

Figure 8: Saturation lifetime: CT mode v.s. non-CT mode

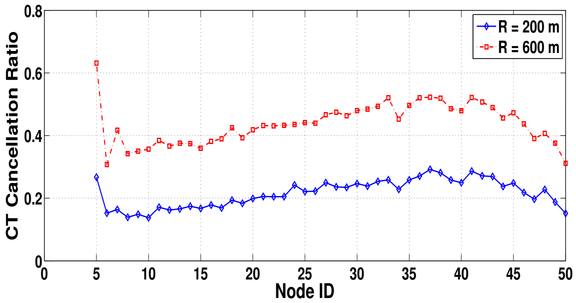

Figure 9: CT cancellation frequency

REFERENCES

[1] J. Li and P. Mohapatra, "An analytical model for the energy hole problem in many-to-one sensor networks," in *Proc. IEEE VTC 2005*.

[2] J. W. Jung and M. A. Ingram, "Residual-Energy-Activated Cooperative Transmission (REACT) to Avoid the Energy Hole," in *Proc. IEEE Int Communications Workshops (ICC) Conf*, 2010, pp. 1–5.

[3] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proc. IEEE INFOCOM 2002*.

[4] "IEEE Std 802.11b-1999 (R2003)," *LAN/MAN Standards Committee of the IEEE Computer Society, IEEE-SA Standards Board*, 2003.

[5] Y. Sun, S. Du, O. Gurewitz, and D. B. Johnson, "DW-MAC: a low latency, energy efficient demand-wakeup MAC protocol for wireless sensor networks," in *Proc. MobiHoc 2008*. New York, NY, USA: ACM.

[6] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks," in *Proc. SenSys 2006*. New York, NY, USA: ACM, pp. 307–320.

[7] F. Tong, W. Tang, R. Xie, L. Shu, and Y.-C. Kim, "P-MAC: A Cross-Layer Duty Cycle MAC Protocol Towards Pipelining for Wireless Sensor Networks," in *IEEE ICC, 2011*, June, pp. 1 –5.

[8] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Trans. on Information Theory*, vol. 50, no. 12, pp. 3062–3080, 2004.

[9] P. Liu, Z. Tao, S. Narayanan, T. Korakis, and S. S. Panwar, "CoopMAC: A Cooperative MAC for Wireless LANs," *IEEE JSAC*, vol. 25, no. 2, pp. 340–354, 2007.

[10] S. Moh and C. Yu, "A Cooperative Diversity-Based Robust MAC Protocol in Wireless Ad Hoc Networks," *IEEE Trans. on Parallel and Distributed Systems*, vol. 22, no. 3, pp. 353–363, 2011.

[11] Y. Zhou, J. Liu, L. Zheng, C. Zhai, and H. Chen, "Link-Utility-Based cooperative MAC protocol for Wireless Multi-hop Networks," *IEEE Trans. on Wireless Communications*, vol. 10, no. 3, pp. 995–1005, 2011.

[12] J. Lin and M. A. Ingram, "SCT-MAC: A Scheduling Duty Cycle MAC protocol for Cooperative Wireless Sensor Networks," *in IEEE International Conference on Communications (ICC)*, June 2012.

[13] H. Jiao, M. Ingram, and F. Li, "A cooperative lifetime extension MAC protocol in duty cycle enabled wireless sensor networks," in *MILCOM 2011*, pp. 896 –901.

[14] G. Anastasi, A. Falchi, A. Passarella, M. Conti, and E. Gregori, "Performance measurements of motes sensor networks," in *Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, ser. MSWiM '04. New York, NY, USA: ACM, 2004, pp. 174–181.

[15] K. Fall, "A delay-tolerant network architecture for challenged internets," ser. in Proc. SIGCOMM '03. New York, NY, USA: ACM, 2003.

[16] T. Rappaport, *Wireless Communications: Principles and Practice*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.

[17] G. Bianchi, "Performance analysis of the ieee 802.11 distributed coordination function," *Selected Areas in Communications, IEEE Journal on*, vol. 18, no. 3, pp. 535 –547, Mar 2000.