2010 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing

OCO: a Multi-channel MAC Protocol with Opportunistic Cooperation for Wireless Sensor Networks

Jinbao Li, Desheng Zhang, Longjiang Guo
School of Computer Science and Technology, Heilongjiang University, Harbin, Heilongjiang, China, 150080
Key Laboratory of Database and Parallel Computing of Heilongjiang Province, Harbin, Heilongjiang, China, 150080
longjiangguo@gmail.com, jbli@hlju.edu.cn, zhang@ieee.org

***Abstract*—To handle the triple hidden terminal problems, this paper proposes OCO, an asynchronous multi-channel MAC protocol with opportunistic cooperation for wireless sensor networks. By adopting opportunistic cooperation, OCO effectively alleviates, if not eliminates, the triple hidden terminal problems. More importantly, OCO is fully distributed with no requirements of time synchronization or multi-radio scheme, so it is easy to be implemented on the real sensor nodes. Via the theoretical analysis, the opportunistic probability that a node cooperates with its neighbor is obtained. To validate the effectiveness of opportunistic cooperation, extensive simulations and real testbed experiments were conducted. The simulation and experimental results show that when the number of channels is large or the network loads are heavy, OCO improves energy efficiency and throughput significantly compared with other works in the literature.**

I. INTRODUCTION

Emerging as one of the dominant technology trends, wireless sensor networks (WSNs) have a wide range of potential applications [1]. Recently, some multi-channel MAC protocols (mcMAC), *e.g.*, MMSN [2], Y-MAC [3], and PMC [4], have been proposed to support these applications via parallel transmissions. The mcMACs have several advantages. *First*, because generally mcMACs employ one channel to send control information and multiple channels to send data, the overall channel utilization increases. *Second*, multiple node-pairs can simultaneously start communications on different orthogonal channels, which can achieve higher network throughput and shorter latency. *Third*, mcMACs involve no extra hardware (*i.e.*, multi-radio) cost because the IEEE 802.11 standard provides 12 channels, and MICA2 sensor motes support more than 50 channels.

An mcMAC usually consists of channel selection and media access components. Channel selection decides how to select idle channels for all the nodes efficiently in order to optimize the network performance, while media access decides when and how all the nodes access channels that have been selected for them to avoid packet collisions. Existing media access schemes generally fall in two basic categories: Time Division Multiple Access (TDMA) and Carrier Sense Multiple Access (CSMA). In TDMA [2][3][5][6], time is divided up into discrete time slots, which have equal or unequal lengths based on different schemes. The time slots are selected by all nodes in the network where nodes are scheduled to send or to receive packets based on different schemes. However, TDMA requires time synchronization, which involves considerable overhead. Whereas, in CSMA [4][12][13][14], all the nodes poll channels and initiate a handshake with the receivers when they have packets to send. This scheme is a natural way for nodes to access shared media. Nevertheless, CSMA will bring more collisions than TDMA, because all the nodes in the network may begin to transmit packets simultaneously due to their contention nature.

According to how frequently channel selection is performed, channel selection can be generally classified as static and dynamic. Under static schemes [2], every node chooses its own dedicated channel to receive data, and switches to other channels to send data. To avoid collisions, static schemes have to guarantee to select different channels for different nodes within two-hop neighborhood, so the number of channels required in static schemes is at least as large as the number of two-hop nodes. To overcome this drawback, dynamic schemes [4][5] select channels for communicating node-pairs on demand. Therefore, channels are occupied by busy node-pairs only. Thus, the number of channels required is at most half of the number of two-hop nodes plus the number of Control Channel (**CC**), which is typically set to *one* and is used to reserve Data Channels (**DC**).

Dynamic channel selection and CSMA with duty cycling are jointly considered as suitable schemes for WSNs because of three reasons as follows. *First*, dynamic channel selection schemes require a less number of channels than static schemes [4]. *Second*, CSMA involves no overhead of time synchronization required by TDMA. *Third*, duty cycling is a scheme to solve the idle listening problem in WSNs, which is considered as one of the largest sources of energy waste in WSNs [1]. However, these combined schemes sometimes fail to offer satisfactory performance since Triple Hidden Terminals (THT) undermine network performance, which are defined in Section III.

In this paper, aiming at addressing THT, an asynchronous mcMAC protocol with Opportunistic CoOperation is proposed, called OCO. Involving no overhead of time synchronization and multi-radio, OCO is tailored to handle THT. The key novelty of this work is that all the neighbors of a transmitting node-pair can opportunistically send cooperative packets based on an opportunistic probability to invalidate the channel selection made by this node-pair, when they assume that the node-pair selects a busy channel for transmission. Moreover, the opportunistic probability is consistently varying according to network parameters. The contributions of this work are as follows.

(1) This paper makes the first attempt to apply an opportunistic cooperation scheme in a duty cycle based multi-channel MAC protocol for handling triple hidden terminals problems.

(2) This paper analyzes OCO's performance, and obtains the opportunistic probability by correlating it with the real time network parameters. No previous work gives such an analysis.

(3) The extensive simulation results show that compared with the other four protocols, OCO achieves 16% to 137% more throughput ratios. OCO also has 7% to 19% better energy efficiency ratios. Furthermore, OCO is also implemented in a real testbed. The experimental results show that OCO achieves 27% to 40% more throughput ratios than compared schemes.

978-0-7695-4322-2/10 $26.00 © 2010 IEEE
DOI 10.1109/EUC.2010.45

Authorized licensed use limited to: Rutgers University. Downloaded on July 23,2020 at 21:46:43 UTC from IEEE Xplore. Restrictions apply.

II. RELATED WORK

In this section, related mcMACs are surveyed from two categories: *synchronous* and *asynchronous*, respectively.

A. Synchronous mcMACs

Zhou *et al.* [2] proposed MMSN which is the first mcMAC that takes into account the restrictions imposed by WSNs. Senders in MMSN switch their current channels to channels of intended receivers at the beginning of every slot when they have packets to send. Salajegheh *et al.* [6] proposed HyMAC for WSNs where the communication period consists of a number of frames, which are divided up into scheduled slots and contention slots. The base station allocates specific time slots and channels to all the nodes for communication. Jovanovic *et al.*[7] proposed TFMAC for WSNs, which works similarly with HyMAC except that the schedules are made by all the nodes rather than the base station. Kim *et al.* [3] proposed Y-MAC for WSNs where time is divided up into several fixed-length frames. The frames are composed of a broadcast period and a unicast period. The difference between Y-MAC and above mcMACs is that Y-MAC schedules the receivers rather than the senders to achieve low energy consumption to extend the lifetime of WSNs.

So *et al.* [5] proposed MMAC for *ad hoc* networks by dividing up time into multiple slots, where all the nodes exchange control information on the CC for reservations of DCs at the front of each slot and switch to DCs for data communication at the rest of the slot. Chen *et al.* [8] proposed MAP for *ad hoc* networks. MAP works in the same way to MMAC but has variable-size data time slots, so it avoids the problem that data slot has to be set according to the maximum data packet size. Tzamaloukas *et al.* [9] proposed CHAT for *ad hoc* networks using channel hopping scheme. Under CHAT, all the idle nodes switch among all channels using a common hopping sequence. Moreover, both the sender and the receiver will stop hopping when they are aware of the fact that they have to communicate with each other. Bahl *et al.* [10] proposed SSCH for *ad hoc* networks, which works in a different way to CHAT by adopting multiple hopping sequences for different nodes. In SSCH, a data communication starts when two nodes hop on the same channel. Tzamaloukas *et al.* [11] proposed RICH-DP based on channel hopping for wireless networks, which differentiates itself with a receiver-initiated collision avoidance scheme.

To sum up, above studies design protocols by time synchronization where let all control information or data be sent in some predetermined slots and channels. For larger scale WSNs, however, synchronization itself remains an open issue that is not completely solved on low cost sensor nodes with cheap faulty clocks. One solution is to send SYNC packets periodically, but these SYNC packets may induce considerable overhead, which consumes more energy and makes channels more crowded.

B. Asynchronous mcMACs

Wu *et al.* [12] proposed DCA for *ad hoc* networks where the node uses two radios, one for control information exchanging and the other for data communication. Adya *et al.* [13] proposed MUP for wireless networks. MUP employs two radios like DCA, but it allows both radios to send control information and data interchangeably. Jain *et al.* [14] proposed RBCS for wireless networks with a dedicated radio for control information exchanges. However, this protocol utilizes a receiver-based channel selection scheme via SNR comparisons at receivers. Nasipuri *et al.* [15] proposed a multi-radio MAC protocol for wireless networks. It distinguishes itself by a soft channel reservation scheme as it gives the preference to the channel that was used for the last successful communication.

Above four protocols are based on multi-radio scheme. Exploiting multi-radio can simplify the design of protocols by dedicating one radio on the CC to consistently overhear the control information exchanging. Nevertheless, multi-radio schemes lead to not only larger node size but also more potential energy consumption. More importantly, increasing hardware cost makes the multi-radio schemes unrealistic for large scale WSNs.

Luo *et al.* [16] exploited Distributed Information SHaring mechanism (DISH) and proposed CAM-MAC for *ad hoc* networks. In CAM-MAC, when a node-pair performs a channel reservation on the CC, all the neighbors may send cooperative packets to invalidate the reservation if they aware of the fact that the selected DC or the receiver is unavailable. Luo *et al.* [17] proposed ALTU based on altruistic cooperation, which introduces some specialized nodes called altruists whose only role is to acquire and to share channel usage information.

These two mcMACs are based on DISH. Nevertheless, in every channel reservation, all the idle neighbors of the sender and the receiver will send packets for invalidation if they assume that this reservation is invalid. Therefore, this scheme involves more packet transmission than necessary and easily results in cooperative packet collisions, since many cooperative packets may be sent simultaneously. Thereby, this scheme will consume considerable energy under large-scale WSNs context.

Le *et al.* [4] proposed PMC which utilizes a control theory approach to dynamically add available channel in a distributed method. In PMC, nodes work on current available channels by CSMA, and decide whether to switch to the next available channel based on certain parameters, which vary with channel utilization from time to time. However, computing methods of these parameters need further discussion. Sun *et al.* [18] proposed RI-MAC which is a receiver-initiated MAC protocol for WSNs. It attempts to minimize the time that a sender and the receiver occupy the wireless medium to find a rendezvous time for exchanging data. Wu *et al.* [19] proposed TMCP which is a multi-channel protocol that does not require time synchronization among nodes. However, this protocol is more like a topology control protocol than a MAC protocol. Zhou *et al.* [20] proposed CUMAC using cooperation for underwater WSNs, but it requires an extra hardware, *i.e.*, tone device, on each node to notify collisions, which increases the cost of WSNs.

C. Summary

All the mcMACs mentioned have at least one limitation below: multi-radio requirement; fine-granular time synchronization; considerable cooperation cost. However, the design of OCO avoids all these limitations. OCO only uses one single radio and is fully asynchronous. Even though OCO is based on cooperation either, its opportunistic cooperation scheme greatly reduces the probability that multiple neighbors simultaneously send cooperative packets, and thus conserves more energy to extending the lifetime of WSNs.

Authorized licensed use limited to: Rutgers University. Downloaded on July 23,2020 at 21:46:43 UTC from IEEE Xplore. Restrictions apply.

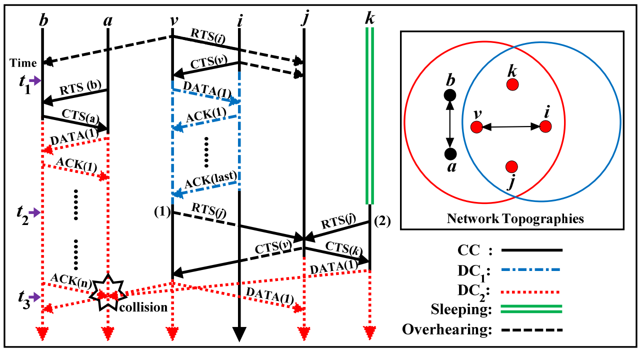


Fig. 1. The illustration of THT

III. DEFINITION OF TRIPLE HIDDEN TERMINALS

This section gives the formal definition of triple hidden terminals, and then explains the reason that causes them.

Triple hidden terminals include three kinds of hidden terminal as follows: **(1)** the *multi-hop hidden terminal* which is the traditional hidden terminal in *multi-hop* networks; **(2)** the *multi-channel hidden terminal* which is a new kind of hidden terminals in *multi-channel* networks [5]; **(3)** the *sleep hidden terminal* which is the latest kind of hidden terminals defined by this paper in *duty cycle based* networks. For multi-hop, multi-channel and duty cycling WSNs together, it will suffer from all the triple hidden terminal problems.

An illustration of THT is given in Fig.1, which involves one CC and two DCs. Node a, b, v, i and j are awake and k is sleeping. When v has data for i, v randomly selects an idle DC such as DC$_1$ and puts the reservation information (*e.g.*, who will occupy which channel for how long) into a RTS and sends to i on the CC. Then, i sends a CTS back to v to confirm this RTS. Next, v and i switch their channels to DC$_1$ around time t_1. The awake neighbors of v and i (*e.g.*, a, b and j) update their channel usage information by overhearing on the CC; whereas, the sleeping neighbors (*e.g.*, k) still assume that DC$_1$ is idle. During $(t_1,\ t_2)$, a has data for b. a randomly selects an idle DC such as DC$_2$ and then switches to DC$_2$ with b after a reservation. Because v and i as well as k are not overhearing on CC during $(t_1,\ t_2)$, v, i and k still assume that DC$_2$ is idle. Around time t_3, two situations will cause packet collisions at a or b. **(1)** When v finishes sending data to i, v has data for j. If v also selects DC$_2$ that a and b are still occupying, then a collision will happen. In this case, v is called the *multi-channel hidden terminal* of a and b. **(2)** When k wakes up, k has data for j. If k also selects DC$_2$ that a and b are still occupying, then a collision will happen as well. In this case, k is called the *sleep hidden terminal* of a and b.

In current studies, the multi-hop hidden terminals can be easily handled via a RTS & CTS handshake scheme. Moreover, the multi-channel hidden terminals can be solved by a fine-granular time synchronization [5], multi-radio scheme [12] and DISH [16], even though as mentioned before all these schemes have some limitations. However, to the best of authors' knowledge, there is no scheme to handle the sleep hidden terminals in the current literatures. Motivate by these issues, this paper tackle THT from a new aspect, *i.e.*, *opportunistic cooperation*.

IV. DESIGN OF OCO

Assumptions are made as follows. (1) Wireless bandwidth is orthogonally divided into one dedicated CC for control packet exchanging and K DCs for communication. (2) Each sensor is equipped with single switchable half-duplex radio.

A. Overview of OCO

OCO is a dynamic duty cycle based asynchronous mcMAC with opportunistic cooperation. OCO utilizes a sender centric coordination to wake up the receiver by multiple RTS packets, according to the predetermined duty cycle. In addition, each idle node periodically and independently turns its radio on and off based on its own duty cycle. This independent sleeping schedule of each node reflects the asynchronization of OCO. Furthermore, in OCO, every node opportunistically cooperates with the transmitting neighbor to select an idle DC for it.

In OCO, all the nodes take five actions as follows. **Overhearing:** When an active node is idle, it monitors the CC to overhear control information exchanging to update its Channel Usage Information (CUI) for next channel reservation. **Reserving:** When it has packets to send, it uses a handshake scheme with the receiver on the CC to negotiate a DC for data communication. **Communicating:** After reserving, this node and the receiver employ media access scheme for communication on the DC they reserved. **Opportunistic Cooperation:** When a node overhears its neighbors reserving a busy DC, it computes an opportunistic probability p, and then it sends a cooperative packet with p to inform its neighbor that it should select again, or it continues to overhear with $1-p$. **Duty cycling:** When being idle for a certain length of time decided by the duty cycle, the node turns off its radio and enters sleeping period for a certain length of time, which also decided by the duty cycle.

The reason that a node opportunistically cooperates with its neighbor with probability p is to make the expected number of cooperative nodes is equal to or less than 1. Therefore, OCO can greatly alleviate, if not completely eliminate, collisions between cooperative packets.

B. Channel Selection of OCO

The channel selection of OCO is a dynamic scheme. When a sender has packets to send, it uses its CUI to obtain the Expected Idle DCs (EIDC). Next, this sender makes these EIDCs into a list, called EIDC List ($\boldsymbol{EIDCL}$), and then sends $EIDCL$ to the receiver with a RTS. When this $EIDCL$ is received, the receiver does the same actions to obtain its $EIDCL$, and computes the intersection of $EIDCL$s, called Final $EIDCL$($\boldsymbol{FEIDCL}$), and then randomly chooses an EIDC from $FEIDCL$, and finally broadcasts the ID number of that EIDC to all the neighbors in an ANnounCement packet ($\boldsymbol{ANC}$). If the neighbors of the receiver overhear this ANC and find that the selected EIDC is actually busy, then its neighbors compute an opportunistic probability p based on the condition of network (address in Section V), and then send a cooperative packet ($\boldsymbol{COP}$) to the receiver with p. A COP includes the CUI of a neighbor of the receiver. Therefore, the receiver can update its $FEIDCL$ to choose a new EIDC, and sends another ANC to make an announcement. When no COP is received, the receiver notifies the sender to switch that EIDC for transmission with a CTS.

Authorized licensed use limited to: Rutgers University. Downloaded on July 23,2020 at 21:46:43 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1: **Media Access of OCO**

```
If (upper layer message coming) { put message into packet buffer queue; }
If (sleeping timer fired){ turn off radio; set up active timer by duty cycle;}
If (active timer fired){ turn on radio; set up sleeping timer by duty cycle;}
If ($CTS$ timer fired) { send the ID number of EIDC selected in $CTS$ to $S$;}
If (sending timer fired){
    check whether $R$ is on the DC by CUI; use CCA to sense the CC;
    If ($R$ is on DC || CC is busy){
        back off for a while and try to send later;}
    Else {obtain $EIDCL$ by CUI; send it in $RTS$ to $R$;}};
If (receiving a packet){
    If (packet is $RTS$){ // as a receiver
        obtain $EIDCL$ by CUI; obtain $FEIDCL$; randomly select an EIDC
        in $FEIDCL$; broadcast ID number of that EIDC in $ANC$ within
        one hop; set up $CTS$ timer;}
    If (packet is $ANC$){ // as a neighbor
        If (EIDC in $ANC$ is busy){
        compute probability $p$; send its CUI in $COP$ to $R$ with $p$;}}
    If (packet is $COP$){ // as a receiver
        obtain a new $EIDCL$ by CUI in $COP$; randomly select an EIDC
        in $FEIDCL$; broadcast ID number of that EIDC in $ANC$ within
        one hop; set up $CTS$ timer;}
    If (packet is $CTS$){ switch to EIDC; send $DATA$s to $R$;}// as a sender
    If (packet is $DATA$){ relay to the upper layer;}// as a receiver};
```

C. Media Access of OCO

Two new kinds of packet are included in OCO, which are $\boldsymbol{ANC}$ (used to announce the EIDC that the receiver tries to use) and $\boldsymbol{COP}$ (used to inform the receiver that it needs to update its $FEIDCL$ for reselecting a new EIDC). The media access of OCO is given in Algorithm 1 where S and R represent a sender and the receiver. In OCO, a node-pair precedes an actually data communications with an opportunistic cooperation ($RTS/ANC/COP/CTS$). The opportunistic cooperation is used to negotiate a EIDC by this node-pair and their neighbors.

D. An Illustration of OCO

An illustration of OCO is shown in Fig.2, which involves one CC and three DCs. Three node-pairs, i.e., AB, CD, and EF, are communicating on DC$_2$, DC$_2$ and DC$_1$, respectively. G is a neighbor of S, and H is a neighbor of R. Both S and R overheard the channel announcements of AB and CD, but missed that of EF due to sleeping. Nevertheless, both G and H overheard the channel announcements of EF. When S has packets for R, three phases must be accomplished as follows.

(1) Handshake Phase$[\boldsymbol{t_0}, \boldsymbol{t_1}]$**:** Via its CUI, S computes $EIDCL$ recording that DC$_1$ and DC$_3$ are idle, and then S sends a RTS with $EIDCL$ to R. When R receives this RTS, R computes its own $EIDCL$, and then computes $FEIDCL$ via their $EIDCL$s, and randomly selects an EIDC, and broadcasts the ID of this EIDC in an ANC .**(2) Opportunistic Cooperative Phase**$[\boldsymbol{t_1}, \boldsymbol{t_2}]$**:** Assume DC$_1$ is the EIDC selected by R, and then H and G receive this ANC correctly. Since DC$_1$ is occupied by EF, both H and G should send a COP to R with opportunistic probability p they computed. Assume H actually sends a COP and G does not. Therefore, R continues to select a new EIDC, say DC$_3$, and sends its ID in a new ANC. After a length of time, no COP is received by R, and then R sends the ID of DC$_3$ in a CTS to S. Neighbors of R update their CUI via this CTS. **(3) Data Communication Phase**$[\boldsymbol{t_2}, \boldsymbol{t_3}]$**:** After S receives this CTS, S and R switch to DC$_3$ and communicate with each other.

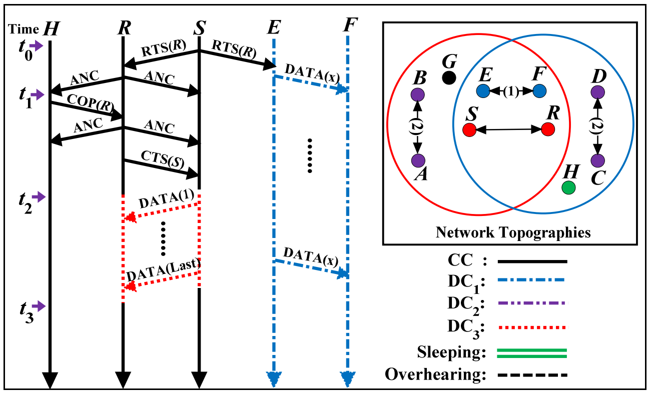


Fig. 2. The Illustration Of OCO

V. THEORETICAL ANALYSIS

In this section, we make a theoretical analysis for the performance of OCO. In particular, we compute the opportunistic probability with which all the neighbors of a receiver that selected a busy DC can send a cooperative packet, but the expected number of cooperative packets sent by all the neighbors is equal to or less than 1. Therefore, OCO can greatly alleviate collisions between cooperative packets, and thus can conserve more energy to extend the lifetime of WSNs.

Assume that the packet arrival process is Poisson arrival process. Let T_0 be a sufficiently long time. During T_0, the total Number of Arrival Messages at each node, denoted as NAM, is given by

$$NAM = \lambda T_0 / AVG, \tag{1}$$

where λ is the average data packet arrival rate at a node, and AVG is the average number of packets in a message. Therefore, the total Time that a node Sends all these Messages on DCs, denoted as TSM, is given by

$$TSM = T_{\mathrm{DC}} \cdot \lambda T_0 / AVG, \tag{2}$$

where T_{DC} is the average duration length of message communications on a DC, which consists of multiple data packet transmissions.

Whereas, approximately, the Total time that a node Receives all these Messages, denoted as TRM, is given by

$$TRM = p_{rcv} \cdot (1 - p_{cc} - p_{slp}) \cdot T_0, \tag{3}$$

where p_{rcv} is the probability that a node switches to a DC as a receiver, p_{cc} is the probability that a node is on the CC at an arbitrary time, and p_{slp} is the probability that a node is sleeping at an arbitrary time.

In the long run, the total time that a node sends messages is supposed to equal the total time that a node receives messages, when the networks is stable, *i.e.*, after sufficiently long time. Therefore, via (2) and (3), we have

$$T_{\mathrm{DC}} \cdot \lambda T_0 / AVG = p_{rcv} \cdot (1 - p_{cc} - p_{slp}) \cdot T_0. \tag{4}$$

Assuming that $p_{rcv} = 1/2$ in the long run, we have

$$p_{cc} = 1 - p_{slp} - 2\lambda T_{DC} / AVG. \tag{5}$$

Authorized licensed use limited to: Rutgers University. Downloaded on July 23,2020 at 21:46:43 UTC from IEEE Xplore. Restrictions apply.

The duty cycle, denoted by q, is given by the design of protocols, and is defined as the idle listening time on the CC (denoted by T_{idle}) over the idle time on the CC plus sleeping time (denoted by T_{sleep}), *i.e.*,

$$q = \frac{T_{idle}}{T_{idle} + T_{sleep}}. \tag{6}$$

We transform (6) into

$$q = \frac{T_{idle}/T_0}{T_{idle}/T_0 + T_{sleep}/T_0}. \tag{7}$$

Because we assumed T_0 is a sufficiently long time, we have

$$q \approx \frac{p_{idle}}{p_{idle} + p_{slp}}. \tag{8}$$

where p_{idle} is the probability that a node is idle on the CC at an arbitrary time. Moreover, via (8), we have

$$q = \frac{1}{1 + p_{slp}/p_{idle}} < \frac{1}{p_{slp}/p_{idle}} = \frac{p_{idle}}{p_{slp}}. \tag{9}$$

Because the situation that a node is on the CC includes two cases (*i.e.*, on one hand, it is idle on the CC; on the other hand, it is exchanging control packets on the CC), we have

$$p_{idle} < p_{cc}. \tag{10}$$

Therefore, via (9) and (10), we have

$$q < \frac{p_{idle}}{p_{slp}} < \frac{p_{cc}}{p_{slp}}. \tag{11}$$

Via (5) and (11), we have

$$p_{cc} > \frac{1 - 2\lambda T_{DC}/AVG}{1 + 1/q}. \tag{12}$$

Therefore, in the long run, the Expected Number of any node's neighbors that on the CC, denoted as ENC, is given by

$$ENC = N \cdot p_{cc} > N \cdot \frac{1 - 2\lambda T_{DC}/AVG}{1 + 1/q}. \tag{13}$$

where N is the average number of neighbors of a receiver, *i.e.*, network density.

Therefore, the upper bound of opportunistic probability p, denoted as p^*, with which all the neighbors of a node on the CC may send a cooperative packet is given by

$$p^* = \min(1, \frac{1}{ENC}) < \min(1, \frac{1 + 1/q}{N \cdot (1 - 2\lambda T_{DC}/AVG)}). \tag{14}$$

During every cooperative phase in Section IV.D, when a receiver selected a busy DC, all the neighbors of this receiver on the CC *uniformly* choose p in interval $(0, p^*)$, and opportunistically send an cooperative packets with p or continue to overhear with $1 - p$. Therefore, the expected number of cooperative packets sent by all the neighbors with opportunistic probability is equal to or less than 1. Note that N and q can be set according to the network deployment; λ can be obtained by a packet counter in the MAC layer; and T_{DC}as well as AVG can be real-time estimated by a progressive weighted method. Therefore, p is dynamically adaptive according the network parameters.

VI. PERFORMANCE EVALUATION

We conduct both simulation and real testbed experiments to examine the performance of OCO, respectively.

A. Simulation Experiments

We built a simulator with C++ for performance comparison where 289 nodes, whose radio communication ranges are set to 40m, are uniformly deployed in a square area of size 200m ×200m with a node density of 38 (*i.e.*, a node that is not at the edge of the network has 37 neighbors). The traffic model is many to many. The payload size is 32 Bytes, and the channel bandwidth is 250 Kbps. The duty cycle is 50%.

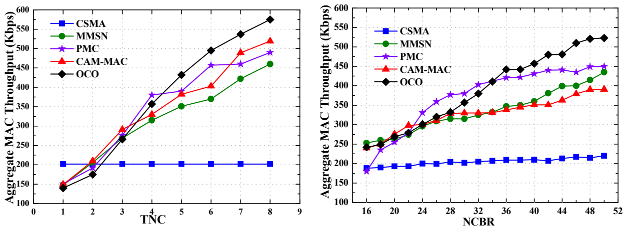
To investigate values of opportunistic cooperation, OCO is compared with 4 protocols: (1) CSMA\CA that is a classic single channel MAC protocol; (2) MMSN [2] that is a typical synchronous mcMAC with a static channel selection for WSNs; (3) PMC [4] that is an asynchronous mcMAC for WSNs with a dynamic channel selection; (4) CAM-MAC [16] that is a synchronous mcMAC for *ad hoc* networks with DISH.

Four groups of simulations are conducted to examine four metrics as follows: aggregate throughput, packet delivery ratio, communication latency and energy consumption. Different total number of channels and loads are considered. The Total Number of Channels (TNC) involves the CC and all the DCs, and the network loads are varied by the change of the Number of CBR (Constant Bit Rate, NCBR) streams in the networks.

***1)* Evaluation on throughput:** The aggregate throughput is computed as the total amount of all useful data packets successful delivered via the MAC layer in the networks per unit time. Intuitively, OCO would have small throughput than other protocols due to its duty cycle scheme, but OCO is expected to increase throughput by efficiently solving THT.

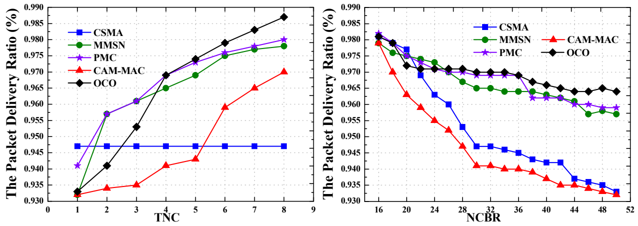
The effect of the total number of channels on throughput is shown in Fig.3 (a). 30 CBR streams are used in this simulation. Compared with others, OCO has lower throughput when the total number of channels is smaller than 4. Besides the duty cycling, this is also due to the fact that the nodes in OCO perform opportunistic cooperation where all the neighbors of the receivers cooperates with opportunistic probability. This opportunistic cooperation scheme will pay a considerable cost if the total number of channels is small. However, when more channels are available, OCO, CAM-MAC and PMC allow more nodes to communicate on different DCs simultaneously. This is because they employ the dynamic channel selections, and thus outperform CSMA\CA with single channel scheme and MMSN with the static selection. However, when the total number of channels becomes larger, OCO performs progressively better than CAM-MAC and PMC. This is because CAM-MAC suffers from the collisions of cooperative packets and PMC suffers from THT; whereas OCO greatly alleviates the collisions of cooperative packet and tackles THT with opportunistic cooperation effectively. Note that when the total number of channel is larger than 6, the gap on throughput between CAM-MAC and OCO is also becoming larger, which suggests that OCO is suitable for the networks with more channels.

Authorized licensed use limited to: Rutgers University. Downloaded on July 23,2020 at 21:46:43 UTC from IEEE Xplore. Restrictions apply.



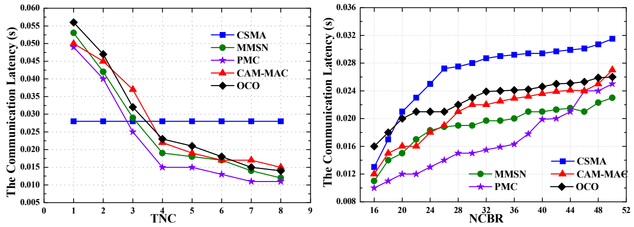
(a)Throughput vs. Number of Channels (b) Throughput vs. Loads

Fig. 3. Throughput evaluation



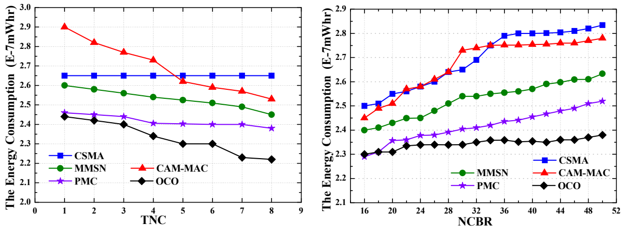
(a) PDR vs. Number of Channels (b) PDR vs. Loads

Fig. 4. Packet delivery ratio evaluation



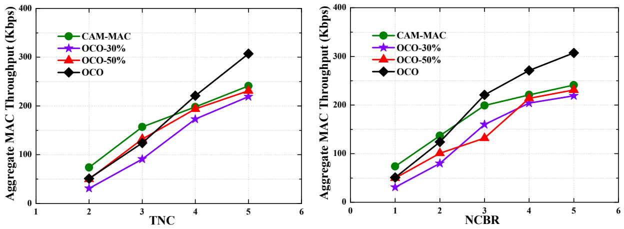
(a) Latency vs. Number of Channels (b) Latency vs. Loads

Fig. 5. Latency evaluation



(a) Energy vs. Number of Channels (b) Energy vs. Loads

Fig. 6. Energy consumption evaluation



(a)Throughput vs. Number of Channels (b) Throughput vs. Loads

Fig. 7. Testbed evaluation on throughput

The effect of the network loads on throughput is shown in Fig.3 (b). The total number of channels is set to 4. It is observed that the throughputs of all the protocols rise with the number of CBR streams. This is because if more node-pairs are involved in data communications, more parallel transmissions will occur on the DCs. Under light loads, *i.e.* the number of CBR streams is smaller than 28, OCO underperforms other protocols. However, the results show that under heavy loads, *i.e.* the number of CBR streams is larger than 34, OCO performs progressively better than others, which shows that OCO significantly benefits from the opportunistic cooperation when the degree of THT increases with the network loads, even though OCO is still duty cycling.

2) **Evaluation on packet delivery ratio:** The Packet Delivery Ratio (PDR) is computed as the ratio of the total number of packets that MAC layer successful delivered over the total number of packets that the upper-layer requests MAC layer to deliver. By avoiding that more than one neighbor cooperates with the receiver on the CC, OCO is expected to increase packet delivery ratio by sending data packets before they are dropped due to exceeding packet's lifetime.

The effect of the total number of channels on packet delivery ratio is shown in Fig.4 (a). CBR streams are set to 30 in this simulation. The results show that the packet delivery ratio increases with the rise of the total number of channels. When the total number of channels is smaller than 4, MMSN and PMC achieve better performances than CAM-MAC and OCO. One possible reason is that schemes of CAM-MAC and OCO to handle THT undermine the packet delivery ratio. However, when the total number of channels is larger than 5, OCO performs better than other protocols due to the fact that OCO effectively copes with THT that becomes more severe when the total number of channels becomes larger. The packet delivery ratio of CAM-MAC is the worst in all the mcMACs, which is due to the fact that the senders under CAM-MAC drop many packets, because the collisions on the CC becomes more serious, and thus the senders cannot reserve a DC in time. Note that when the total number of channel is larger than 4, the gap on packet delivery radio between OCO and PMC is becoming larger. This suggests that OCO is more suitable than PMC in the networks with more available channels.

The effect of the network loads on packet delivery ratio is shown in Fig.4 (b). The total number of channels is set to 4. In Fig.4 (b), all packet delivery ratios generally drop when the network loads are heavier except that of OCO. When the number of CBR streams is larger than 34, the packet delivery ratio maintains steady. This is primarily due to the fact that under opportunistic cooperation of OCO, node-pairs more likely find an idle DC for communication in time before the packets are dropped by the sender due to exceeding their lifetime. Furthermore, it is observed that with the rise of the number of CBR streams, the packet delivery ratio of OCO outperforms that of CAM-MAC significantly, which vaildates that the opportunistic cooperation outweighs DISH scheme in the network with more loads.

Authorized licensed use limited to: Rutgers University. Downloaded on July 23,2020 at 21:46:43 UTC from IEEE Xplore. Restrictions apply.

3) **Evaluation on latency**: The communication latency reflects the time delay from the time instance that a data packet is delivered to the MAC layer from the upper layer to the time instance that this packet is actually sent out. Intuitively, OCO will bring longer latency than others due to its duty cycling scheme and the multiple opportunisic cooperative handshake on the CC, but the results show that the difference between these protocols is negligible when THT becomes more serious.

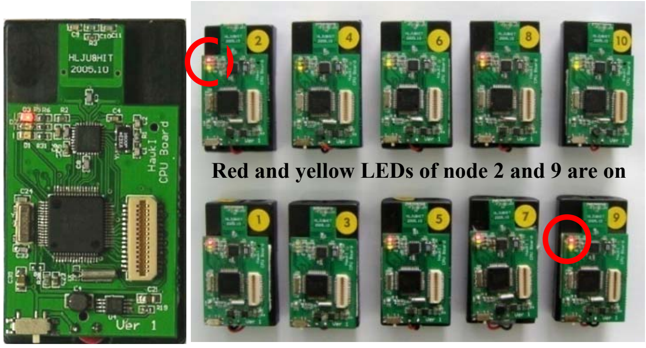
The effect of the total number of channels on latency is shown in Fig.5 (a). The number of CBR streams is set to 30 in this simulation. The results show that compared with others, OCO has a similar or larger latency when the total number of channels is smaller than 3. This is because that OCO is duty cycling to achieve energy efficiency and the opportunisic cooperation scheme involves more handshake control packets. However, when the total number of channel increasingly steps up, the difference on latency becomes negligible. This is because that other protocols suffer the retransmission problem resulted from THT.

The effect of the network loads on latency is shown in Fig.5 (b). The total number of channels is set to 4. It is observed that when the network loads are light, *i.e.*, the number of CBR streams is smaller than 30, OCO has a larger latency than most of other protocols due to its opportunistic cooperation. However, when THT becomes severer as the network loads are heavier, the gap between OCO and other protocols on latency becomes narrower, since OCO effectively addresses THT.

4) **Evaluation on Energy Consumption:** In this study, the energy consumption for all the protocols is computed as the energy consumed to successfully deliver a useful data byte. OCO is expected to achieve energy efficiency via its opportunistic cooperation and avoiding the energy consumption for time synchronization and the retransmission caused by THT.

The effect of the total number of channels on energy consumption is shown in Fig.6 (a). The number of CBR streams is set to 30. The results show that the energy consumptions of all the protocols decrease with the rise of the total number of channels, but OCO outperforms others all the time. This indicates that OCO can conserve more energy to prolong the lifetime of WSNs by avoiding time synchronization of MMSN and continuously channel switching of PMC. Whereas, CAM-MAC consumes higher energy than OCO all the time due to its collisions of cooperative packets, which undermine many communications when THT is less serious.

The effect of the network loads on energy consumption is shown in Fig.6 (b). The total number of channels is set to 4. All energy consumptions increase when the loads rise. OCO maintains lower energy consumption when the number of CBR streams is larger than 20. This is because the other protocols suffer from certain problems. MMSN consumes a larger amount of energy to maintain time synchronization among all the nodes; PMC has many collisions on the current channel when the loads are heavy; CAM-MAC seriously suffers from collisions between cooperative packets and reservation packets when more node-pairs communicate simultaneously.



Red and yellow LEDs of node 2 and 9 are on

(a) Hawk node (b) A snapshot in the testbed experiment

Fig. 8. Testbed experiments

B. Testbed Experiments

In this section, to examine the effectiveness of opportunistic cooperation scheme, we introduce the testbed experiment on OCO, and analyze the experimental results.

We built a sensor node platform, Hawk, for our experiments. Several testbed experiments were conducted to evaluate the performance of OCO. Hawk employs μC/OS, where each node is equipped with an nRF905 radio and a MSP430 processor. A hawk node is shown in Fig.8 (a).

For visualization purposes, three LEDs are employed (red, green, and yellow) on every node in the testbed to indicate specific events, *e.g.*, when the red LED is on, it indicates that the node is communicating on a DC. Meanwhile, the green LED and the yellow LED jointly indicate the specific ID number of DCs (a maximum of 2^2=4 DCs can be represented), *e.g.*, in Fig.8 (b) the red and yellow LEDs of node 2 and 9 are on, which indicates they are communicating on DC_3. When the red LED is off and the yellow LED and the green LED are on, it indicates the node is on the CC. When all the three LEDs are off, it indicates the node is sleeping period.

The testbed consists of 10 hawk nodes which are completely connected as shown in Fig.8 (b). In such a case, all the nodes are within the communication range of each other, similar to the experimental deployment in [16]. The size of each packet is 32 Byte, and data transmission rate is 100 Kbps. All the nodes randomly choose a neighbor to initiate a data communication. The experiment was repeated for 10 times. When an experiment is finished, all the nodes send their total number of bytes received during this experiment to a sink node one by one, which is connected to a desktop computer, and thus throughput can be obtained.

Due to the time synchronization of MMSN and the complexity of PMC for parameter computations, only OCO and CAM-MAC were implemented in the testbed, and they were compared with two varieties of OCO. These two varieties are to validate the effect of opportunistic cooperation. First variety is called OCO-30%, which works similarly with OCO except that it utilizes a static cooperation probability of 30%; second variety is called OCO-50%, which utilizes a static cooperation probability of 50%.

Authorized licensed use limited to: Rutgers University. Downloaded on July 23,2020 at 21:46:43 UTC from IEEE Xplore. Restrictions apply.

The effect of the total number of channel on throughput is shown in Fig.7 (a). The number of CBR streams is set to 5. It is observed that CAM-MAC has higher throughput than OCO when the total number of channel is less than 3. This is primarily because CAM-MAC does not have to enable duty cycling, which undermines OCO's throughput. However, OCO achieves better throughput than CAM-MAC does when the total number of channel is larger than 4. An explanation of this improvement is that when more channels are available, THT becomes more serious, and OCO tackles THT with less cost in opportunistic cooperation than CAM-MAC does in DISH. Note that these experimental results are generally consistent with the simulation results in tendency shown in Fig.3 (a), even though the setup of the testbed experiment is different from that of simulations. Moreover, OCO has higher throughput than both OCO-30% and OCO-50%, which further validates the effectiveness of the opportunistic cooperation scheme adopted by OCO.

The effect of loads on throughput is shown in Fig.7 (b). In this testbed experiment, the total number of channels is set to 5. It is observed that all three OCO based protocols have lower throughput than CAM-MAC when network loads are small, *i.e*., the number of CBR streams is less than 3. This is because that when fewer nodes are involved in communication, the opportunistic cooperation of OCO works less efficiently to tackle THT than DISH of CAM-MAC. This is also because that when the network loads are light, the collision on the CC between cooperative packets is rarer. Nevertheless, when more loads are involved in the networks, OCO achieves better throughput than CAM-MAC. This is because when the loads are heavier, OCO avoids the collisions on the CC by opportunistic cooperation. Note that OCO works better than OCO-30% and OCO-50%. More importantly, the gap on the throughput between OCO and them becomes larger when the loads are heavier, which shows that opportunistic cooperation actually improves the throughput of OCO in the networks with heavy loads.

VII. CONCLUSION

The triple hidden terminal problems are major causes of energy wastage in WSNs. To address these problems, in this paper, an asynchronous multi-channel duty cycle based MAC protocol, called OCO, is proposed. OCO exploits opportunistic cooperation scheme to effectively handle the triple hidden terminal problems. Being distributed with no requirements of time synchronization or multi-radio, M-cube is suitable to be employed in large-scale WSNs. By theoretical analysis, the opportunistic probability is obtained, which constantly varies according to the real-time estimations of the protocol parameters to make sure that the expected number of cooperative neighbors is equal to or less than 1. Moreover, extensive simulations were conducted to examine the performance of OCO. The results show that with opportunistic cooperation, OCO can solve the triple hidden terminal problems with a lower cost, and still enable duty cycling at the same time. Thereby, OCO achieves a significant improvement of energy efficiency and other performances as well, especially when the total number of channels and loads increase. We also implemented OCO on a real sensor platform. The testbed results show that opportunistic cooperation actually enables OCO to achieve better throughput.

VIII. ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (61070193, 60803015), the Key Scientific and Technological Research Project of Heilongjiang Province of China (GC09A109), the China Postdoctoral Science Foundation (20080430902), the Science and Technology Innovation Research Project of Harbin for Young Scholar (2008RFQXG 107), the Heilongjiang Postdoctoral Science Foundation (No. LRB08-021), the Science and Technology Key Research of Heilongjiang Educational Committee (1154Z1001), the Heilongjiang University Foundation for Young Scholar, the Innovative Research Project of Heilongjiang University (2010208) and Innovative Laboratory Project of Heilongjiang University (2010046).

REFERENCES

[1] D. Culler, D. Estrin, and M. Srivastava. Overview of Sensor Networks, in *IEEE Computer*, *Special Issue on Sensor Networks*, 2004.

[2] G. Zhou, C. Huang, Ting Yan, Tian He, John A. Stankovic and Tarek F. Abdelzaher. MMSN: Multi-Frequency Media Access Control for Wireless Sensor Networks, in *INFOCOM*, 2006.

[3] Y Kim, H Shin, H. Cha. Y-MAC: An Energy-efficient Multi-Channel MAC Protocol for Dense Wireless Sensor Networks, in *IPSN*, 2008.

[4] H. K. Le, D. Henriksson, *et.al*. A Practical Multi-Channel Media Access Control Protocol for Wireless Sensor Networks, in *IPSN*, 2008.

[5] J. So and N. Vaidya. Multi-Channel MAC for Ad Hoc Networks: Handling Multi-Channel Hidden Terminals Using a Single Transceiver, in *MOBIHOC*, 2004.

[6] M. Salajegheh，H. Soroush, and A. Kalis. HyMAC：Hybrid TDMA/ FDMA Medium Access Control Protocol for Wireless Sensor Networks, in *PIMRC*, 2007.

[7] M. Jovanovic, and G. Djordjevic. TFMAC: Multi-Channel MAC Protocol for Wireless Sensor Networks, in *TELSIKS*, 2007.

[8] J. Chen, S. Sheu, and C. Yang. A New Multichannel Access Protocol for IEEE 802.11 Ad Hoc Wireless LANs, in *PIMRC*, 2003.

[9] A. Tzamaloukas and J.J.Garcia-Luna-Aceves. Channel-Hopping Multiple Access with Packet Trains for Ad Hoc Networks, in *MoMuC*, 2000.

[10] P. Bahl, R. Chandra, and J. Dunagan. SSCH: Slotted Seeded Channel Hopping for Capacity Improvement in IEEE 802.11 Ad-Hoc Wireless Networks, in *MOBICOM*, 2004.

[11] A.Tzamaloukas and J.J. Garcia-Luna-Aceves. A Receiver-Initiated Collision-Avoidance Protocol for Multi-Channel Networks, in *INFOCOM*, 2001.

[12] S.L. Wu, C.Y. Lin, Y.C. Tseng, and J.P. Sheu. A New Multi-Channel MAC Protocol with On-Demand Channel Assignment for Multi-Hop Mobile Ad Hoc Networks, in *ISPAN*, 2000.

[13] A. Adya, P. Bahl, J. Padhye, and A. Wolman. A Multi-Radio Unification Protocol for IEEE 802.11 Wireless Networks, in *BROADNETS*, 2004.

[14] N. Jain, S.R.Das, and A, Nasipuri. A Multichannel CSMA MAC Protocol with Receiver-Based Channel Selection for Multihop Wireless Networks, in *ICCCN*, 2001.

[15] A. Nasipuri, J. Zhuang, and S. R. Das. A Multichannel CSMA MAC Protocol for Multihop Wireless Networks, in *WCNC*, 1999.

[16] T. Luo, M. Motani, and V. Srinivasan. Cooperative Asynchronous Multichannel MAC: Design, Analysis, and Implementation,in *IEEE TMC*, vol 8, No 3, 2009.

[17] T. Luo, M. Motani, and V. Srinivasan. Extended Abstract: Altruistic Cooperation for Energy-Efficient Multi-Channel MAC protocols, in *MOBICOM*, 2007.

[18] Y. Sun, O. Gurewitz and D. Johnson. A Receiver Initiated Asynchronous Duty Cycle MAC Protocol for Dynamic Traffic Loads in Wireless Sensor Networks, in *SenSys*, 2008.

[19] Yafeng Wu, John A. Stankovic, Tian He Jiakang Lu, and Shan Lin. Realistic and Efficient Multi-Channel Communications in Wireless Sensor Networks, in *INFOCOM*, 2008.

[20] Zhong Zhou, Zheng Pengt, Jun-Hong Cui, and Zaihan Jiang l. Handling Triple Hidden Terminal Problems for Multi-Channel MAC in Long-Delay Underwater Sensor Networks, in *INFOCOM*, 2010.

Authorized licensed use limited to: Rutgers University. Downloaded on July 23,2020 at 21:46:43 UTC from IEEE Xplore. Restrictions apply.