

M-cube: A Duty Cycle Based Multi-Channel MAC Protocol with Multiple Channel Reservation for WSNs

Jinbao Li^{1,2}, Desheng Zhang^{1,2}, Longjiang Guo^{1,2}, Shouling Ji³, and Yingshu Li³

¹School of Computer Science and Technology, Heilongjiang University, Harbin, Heilongjiang, China, 150080

²Key Laboratory of Database and Parallel Computing of Heilongjiang Province, Harbin, Heilongjiang, China, 150080

³Department of Computer Science, Georgia State University, Atlanta, GA, USA, 30303

longjiangguo@gmail.com, jbli@hlju.edu.cn, zhang@ieee.org, sji@cs.gsu.edu, yli@cs.gsu.edu

Abstract—In this paper, a duty cycle based multi-channel MAC protocol with multiple channel reservation, called M-cube, is proposed to tackle the triple hidden terminal problems. M-cube can make nodes to choose one actually idle channel from all the expected idle channels. Therefore, M-cube can avoid data packet collisions resulted by the triple hidden terminal problems. By minimizing the lower bound of the average number of times of channel switching in M-cube, the optimal duty cycle is obtained through theoretical analysis. To validate the effectiveness of multiple channel reservation and dynamic optimal duty cycling, extensive simulations and real testbed experiments were conducted. Both the simulation and experiment results show that when the number of channels is large or network loads are heavy, M-cube improves energy efficiency and throughput significantly compared with other works in the literature.

I. INTRODUCTION

Recently, to overcome the drawbacks of single-channel MAC protocols, some multi-channel MAC protocols (mcMAC) have been proposed to improve network performance of Wireless Sensor Networks (WSNs) via parallel transmissions [1, 2, 3]. mcMACs have several advantages as follows. *First*, because generally mcMACs employ one Control Channel (CC) to send control information and multiple Data Channels (DC) to send data, the overall channel utilization is increased. *Second*, mcMACs have higher throughput and shorter latency. *Third*, because current WSNs radios already offer multiple channels [1], mcMACs incur no more multi-radio cost. An mcMAC mainly consists of channel selection and media access functions. Channel selection schemes can be classified as *static* and *dynamic* ones. Media access schemes fall in two categories: *TDMA* and *CSMA*.

Dynamic channel selection and CSMA with duty cycling are considered as suitable schemes for WSNs [3]. However, these combined schemes sometimes fail to offer satisfactory performances due to the Triple Hidden Terminals problems (THT), which includes three kinds of hidden terminal: (1) *multi-hop hidden terminals*; (2) *multi-channel hidden terminals* [5]; (3) *sleep hidden terminals*. For multi-hop, multi-channel and duty cycling WSNs, they will severely suffer from THT. As shown in [4], THT is one of the most primary reasons of energy waste in the multi-channel scenario, which results from the fact that channel usage information may not be timely obtained by all nodes. Therefore, when a node selects an Expected Idle Data Channel (EIDC), this EIDC may be already being used by other nodes. The EIDC that is actually busy is called the Misunderstood Channel (MC).

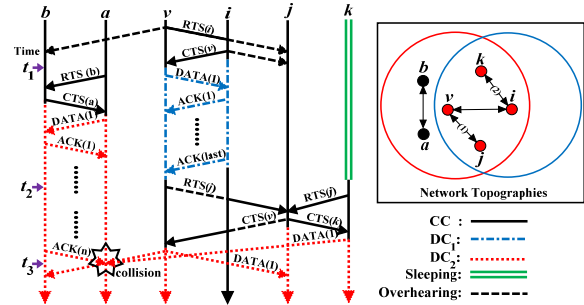


Figure 1. An example of THT

An example of THT is given in Fig.1. It 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 adds the reservation information (e.g., who will occupy which channel for how long) into an RTS and sends to i on the CC. Then, i sends a CTS back to v to confirm the delivery of RTS . Next, v and i switch their channels to DC_1 at 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 another 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. At time t_3 , two situations may 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 being occupied by a and b , a collision may 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 being occupied by a and b , a collision may happen as well. In this case, k is called the **sleep hidden terminal** of a and b .

To solve THT, we propose a dynamic duty cycle based Multi-channel MAC protocol with Multiple channel reservation (M-cube) for heavy loads WSNs. The contributions of this work are as follows. (1) An asynchronous mcMAC, called M-cube, is presented especially for heavy loads WSNs. (2) M-cube's performance is analyzed by probability theory. (3) The extensive simulation results show that compared with the other four protocols, M-cube achieves 6% to 174% more throughput ratios. M-cube also has 6% to 90% better energy efficiency ratios. Furthermore, M-cube is also implemented in a real testbed. The results show that M-cube achieves 23% to 63% more throughput ratios.

II. RELATED WORK

The mcMACs for WSNs are classified into two main categories: *synchronous* and *asynchronous* ones as follows.

A. Synchronous mcMACs for WSNs

Zhou *et al.* [1] proposed MMSN which is the first mcMAC that takes into account the restrictions in WSNs. Salajegheh *et al.* [5] proposed HyMAC where the communication period consists of a number of frames, and the frame is divided into scheduled slots and contention slots. The base station selects channels and specific time slots for all nodes. Jovanovic *et al.* [6] proposed TFMAC where a frame consists of a contention period and a contention-free period that contains some equal sized time slots. TFMAC works similarly with HyMAC except that the schedules are made by all nodes rather than the base station. Kim *et al.* [2] proposed Y-MAC via adding a multi-channel mechanism to Crankshaft [7]. Y-MAC schedules receivers rather than senders to achieve low energy consumption.

B. Asynchronous mcMACs for WSNs

Le *et al.* [3] proposed PMC which involves no time synchronization and utilizes a control theory approach to dynamically add available channels one by one for all nodes in a distributed manner. Wu *et al.* [8] proposed TMCP which is a multi-channel protocol that does not require time synchronization. However, this protocol is more like a topology control protocol rather than a MAC protocol. Ansari *et al.* [9] proposed a spectrum agile mcMAC where all nodes scan all channels and make sure whether there are packets for themselves, which involves the overhead of channel switching.

Unfortunately, all above schemes for WSNs do not consider THT. THT is only *partly* considered by mcMACs for general wireless networks, because in general wireless networks the duty cycle feature is not taken into account. Thereby, the current solutions for THT fail to solve sleep hidden terminals in duty cycle based WSNs. The current solutions for THT can be categorized into three classes: *multi-radio*, *time synchronization* and *distributed information sharing*.

A. Multi-Radio Schemes

Wu *et al.* [10] proposed DCA which uses two radios, one radio for control information exchanging used for channel reservations, and the other radio for data communication. Adya *et al.* [11] proposed MUP which allows both radios to interchangeably send control information and data. Jain *et al.* [12] proposed a protocol with a receiver-based channel selection scheme via SNR comparisons at receivers. Nasipuri *et al.* [13] proposed a multi-radio scheme which distinguishes itself by a soft channel reservation scheme. Multi-radio scheme can solve THT by dedicating a radio on the CC to consistently overhear control information exchanging. However, the requirement of multi-radio leads to not only larger node size but also more potential energy consumption [4], which could result in a shorter network lifetime. Moreover, multi-radio schemes result in high hardware cost, which is unrealistic for most large-scale WSNs.

B. Time Synchronization Schemes

So *et al.* [14] proposed MMAC which partitions time into multiple time slots. In MMAC, all nodes exchange control information on the CC for channel reservations at the beginning of each slot and switch to DCs for data communications in the rest of the slot. Chen *et al.* [15] proposed MAP which works in the same way as MMAC but has variable-length time slots. Compared with the protocols using fixed-length time slots, MAP avoids the problem that the length of a time slot has to be decided according to the maximum data packet size. Tzamaloukas *et al.* [16] proposed CHAT which employs time synchronization in channel hopping scheme. In CHAT, all the idle nodes switch among all the channels using a common hopping sequence. Bahl *et al.* [17] proposed SSCH that is also based on the channel hopping, but SSCH uses multiple hopping sequences for different nodes. These schemes address THT by time synchronization. Most of them send all the control information (*i.e.*, channel reservation information) in some pre-decided time slots. However, time synchronization is still an open problem for low cost sensor nodes with cheap prone to drift clocks [8]. One common solution is to periodically send SYNC packets, but it will consume more energy and make channels more crowded.

C. Distributed Information Sharing Schemes

Luo *et al.* [4] take advantage of Distributed Information SHaring mechanism (DISH) and propose CAM-MAC to address the multi-channel coordination problem. In CAM-MAC, when a communicating node-pair performs a channel reservation on the CC, all neighborhood nodes may send cooperative packets to invalidate the reservation if they are aware of the fact that the selected DC or receiver is unavailable. In addition, Luo *et al.* [18] proposed a multi-channel MAC protocol based on a strategy called altruistic cooperation. This protocol introduces some specialized nodes called altruists in the networks whose only role is to acquire and share channel usage information. Furthermore, Luo *et al.* [19] developed a theoretical treatment of DISH to analytically evaluate the availability of information sharing. Instead of directly analyzing throughput, this study analyzes the availability of information sharing and correlates it with performance metrics including throughput. DISH solves THT by involving more nodes into a channel selection. However, in every channel reservation, all the idle neighbors of the sender and the receiver may send packets for invalidation, if they assume this reservation is invalid. Therefore, it causes more redundant communications and easily results in cooperative packet collisions, because many cooperative packets could be sent simultaneously.

Summary. In this paper, M-cube is proposed for WSNs to tackle THT in a different way from all above-mentioned works. Three important features distinguish M-cube from prior works. *First*, in M-cube nodes are only equipped with one single radio; *second*, M-cube is fully asynchronous; *third*, all communicating node-pairs in M-cube make channel selection decision only based on their own information, *i.e.*, no redundant communications from other nodes are introduced.

III. DESIGN OF M-CUBE

Wireless bandwidth is divided into one dedicated CC for control packet exchanging and K DCs for communication.

A. Overview of M-cube

M-cube is a dynamic duty cycle based asynchronous mcMAC with multiple channel reservation. There are three features of M-cube as follows. (1) M-cube utilizes a sender centric coordination to wake up the receiver by a series of handshake packets (*RTS*), according to the Dynamic Optimal Duty Cycle (**DODC**, discussed in Section IV). Each idle node periodically turns its radio on and off based on its own DODC to conserve energy and to prolong network lifetime of WSNs. (2) The independent sleeping schedule of each node reflects the asynchronization of M-cube. (3) In M-cube, every node-pair reserves multiple EIDCs instead of one. In M-cube, all nodes take four 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 a node has packets to send, it uses a handshake scheme with the receiver on the CC to negotiate a list of common EIDCs for data communication. **Communicating**: After reserving a DC, this node and the receiver employ media access for communication on one of all the DCs they reserved. **Duty cycling**: After being idle for certain duration of time decided by DODC, this node turns off its radio and enters sleep period for certain duration of time, which is also decided by DODC.

B. Channel Selection of M-cube

The popular idea to solve THT is to update the CUI in real-time. This will introduce too much SYNC overhead or hardware cost (e.g., multi-radio). In this study, we tackle THT from a new aspect, i.e., instead of updating the CUI in real-time, we use outdated CUI to take care of THT. The outdated CUI has a property that if the outdated CUI shows that a DC is idle now, then this DC is *probably* idle, whereas if the outdated CUI shows that a DC is busy now, then this DC is *definitely* busy. As shown in Fig.1, this property is resulted by the fact that a node misses some channel reservation information during its sleep period or communications on a DC. We utilize this property to design a channel selection scheme, called multiple channel reservation, in M-cube. When a sender has packets to send, it uses this property to obtain the DCs expected to be idle by its CUI, which could also be busy with a certain probability. Next, this sender makes these EIDCs into a list, called EIDC List (**EIDCL**), and then sends **EIDCL** to the receiver. When this **EIDCL** is received, the receiver performs the same actions to obtain its **EIDCL**, and computes intersection of **EIDCL**s, called Final **EIDCL** (**FEIDCL**), and finally sends **FEIDCL** back to the sender. After that, both the sender and the receiver switch among all the EIDCs in **FEIDCL** based on the random order of channels in **FEIDCL** until they find an actually idle EIDC. When a node-pair finds an actually idle EIDC, a node-pair have to switch back to the CC first and inform all the idle neighbors that they actually use this DC instead of other DCs in **FEIDCL**. Therefore, all these idle neighbors can update their CUI.

Algorithm 1: Media Access of M-cube

```

If (upper layer message coming) { add message into packet buffer queue;}
If (sleeping timer expired) { turn off radio; set up active timer by DODC;}
If (active timer expired ) { turn on radio; set up sleeping timer by DODC;}
If (sending timer expired ) {
  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 tries to send later;}
  Else {obtain AIDCL 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; send it in CTS to  $S$ ;
    While (switch to next DC in FAIDCL){
      monitor this DC for  $2T$ (explain in subsection III.D);
      If (this DC is busy){
        If (node occupying this DC is not a neighbor of  $S$  ) {
          send CSC to inform  $S$  to switch again;}}/
        Else If (receiving the DII packet from  $S$ ){
          send DII on this DC to  $S$ ; switch to the CC;
          inform neighbors which DC it occupied with ANC;
          switch to DC; wait to receive DATA from  $S$ ; send ACK; }
        Else If (receiving CSC){continue;}}}}
      If (packet is CTS){ // as a sender
        While (switch to next DC in FEIDCL){
          monitor this DC for  $T$ ;
          If (this DC is busy){
            If (node occupying this DC is not a neighbor of  $R$ ){
              send CSC to inform  $R$  to switch again;}}
            Else { send DII on this DC to  $R$ ;
              If (receiving DII ) {
                switch to CC; inform neighbors occupied DC with ANC;
                switch to that DC; send DATAs to  $R$ ;
                Else if (receiving CSC ) {continue;}}}}
          If (packet is ANC){ update CUI; } } // as a neighbor
          If (packet is ACK){ send next DATA; } } // as a sender

```

In M-cube, a node-pair reserves multiple EIDCs instead of one is because that if they reserve one EIDC and this DC is actually busy, they have to switch back to the CC and reserve a new EIDC again via another handshake. Moreover, this new EIDC could also be busy. Thereby, reserving only one DC once may result in multiple handshakes on the CC for one message communication consisting of multiple data packet transmissions. These multiple handshakes will compromise the utilization of the CC.

C. Media Access of M-cube

Three new kinds of packets are included in M-cube, which are **CSC** (used to inform a node on a DC that it needs to Continue to Switch Channel among **FEIDCL**), **DII** (used to inform a node on a DC that this DC Is Idle) and **ANC** (used to make an ANnouncement on the CC about the DC a node actually uses). The media access of M-cube is given in Algorithm 1 where S and R represent a sender and the receiver. In M-cube, a node-pair first executes a handshake scheme (**RTS/CTS**) and a channel announcement scheme (**DII/ANC**) before a message communications (**DATAs/ACKs**). Note that senders also are supposed to *receive ACKs* in M-cube, so the DC selected must be idle for both the sender and the receiver. Handshake scheme is used to negotiate a list of EIDCs by this node-pair; while Channel announcement is to select an actually idle DC in **FEIDCL** and to help all their idle neighbors update their CUIs.

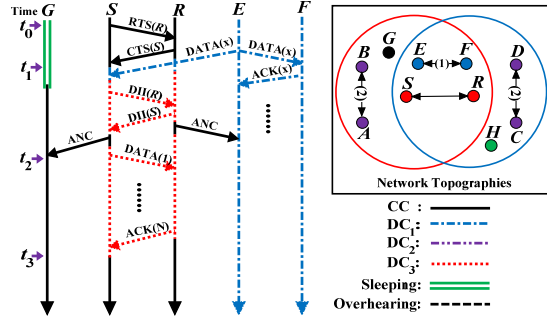


Figure 2. The Illustration Of M-cube

D. An Example of M-cube

Fig.2 describes an execution example of M-cube. There is 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 G and H are sleeping at the beginning, and G wakes up later. Both S and R overheard the channel announcements of AB and CD , but missed that of EF due to sleep or communications. When S has packets for R , three phases must be accomplished as follows.

(1) Handshake Phase $[t_0, t_1]$: Based on its CUI, S computes $EIDCL$ recording that DC_1 and DC_3 are idle, and then S sends an RTS with $EIDCL$ to R . When R receives this RTS , R computes its own $EIDCL$, and then computes $FEIDCL$ via $EIDCL$ s of R and S , and finally sends a CTS with $FEIDCL$ to S . (2) Channel Announcement Phase $[t_1, t_2]$: Assume DC_1 is the first DC in $FEIDCL$, and then both S and R switch to DC_1 and listen for time T and $2T$ where T is decided according to the maximum data packet size. Because DC_1 is occupied by EF , both S and R may receive a packet from E or F , which indicates that DC_1 is busy. Therefore, both S and R continue to switch to DC_3 without sending CSC since they are both aware of the fact that E or F is their common neighbor. After monitoring DC_3 , S and R exchange DII to make sure that DC_3 is idle for both nodes. Then, S and R switch to the CC, and sequentially send the same ANC about this channel selection, which helps their idle neighbors (*e.g.*, G) to update their CUIs. (3) Data Communication Phase $[t_2, t_3]$: S and R switch back to DC_3 and communicate with each other by $DATAs/ACKs$ exchanging. When these exchanging are finished, S and R switch to the CC again and update their CUIs via overhearing the ANC sent on the CC by their communicating neighbors.

IV. THEORETICAL ANALYSIS

In this section, M-cube's performance is theoretically analyzed. In particular, the lower bound of the average numbers of times (denoted as \bar{x}) that a node-pair switches among the DCs in $FAIDCL$ is computed. Represented by the function of the duty cycle q , the value of \bar{x} can basically decide the latency and the energy consumption on channel switching among DCs. Lastly, the optimal duty cycle q^* is obtained, which is defined as the duty cycle that minimizes the lower bound of \bar{x} . The symbols used in the analysis are listed in Table I.

Table I. Symbol list

	Symbol	Meaning
Probabilities	p_{cc}	a node is on the CC at an arbitrary time
	p_{stp}	a node is sleeping at an arbitrary time
	p_{mc}	a MC is created
	p_{rcv}	a node switches to a DC as a ReCeiver
Events	MCC	a MC is Created
	$MAN_{(i)}$	v Misses a ANC from a Neighbor (<i>e.g.</i> , i)
	IIS	i Is a Sender
	$OCC_v(t)$	v is On the CC at time t
	$NSC_v(T_{IA})$	v is Not Sending on the CC during T_{IA}
	$NIC_u(T_{IA})$	u does Not Interfere on the CC during T_{IA}
	$BTC(T_{IA})$	u switches Back To the CC during T_{IA}
	$SLP_u(T_{IA})$	u is SLeePing during T_{IA}
Others	MC	an Expected idle channel that is actually busy
	q, q^*	the duty cycle, the optimal duty cycle
	N	the average number of nodes per r^2 (density)
	λ	the average data packet arrival rate at a node
	AVG	the average number of packets in a message
	T_{WK}	the longest wake time of an idle node
	T_{SP}	the longest sleeping time of an idle node
	T_{AN}	duration of two nodes sending ANC
	T_{DC}	duration of message communications on a DC
	$N_{i_i}, N_{i_v}, N_{v_i}$	i 's neighbor set; $N_{v_i} = N_v \cap N_i$, $N_{v \setminus i} = N_v \setminus N_i \setminus \{i\}$

Let x be the number of times that a node-pair switches among the DCs in $FEIDCL$ until they find an actually idle DC. x is geometrically distributed with parameter p , which represents the probability that an EIDC in $FEIDCL$ is actually busy. We call this busy DC in $FEIDCL$ the Misunderstood Channel (**MC**). The expectation of geometrical variable is

$$\bar{x} = 1/p. \quad (1)$$

The following subsection explains how to derive \bar{x} .

A. Derivation of p

Let v be an arbitrary node in the networks. By the Total Probability Theorem (**TPT**),

$$p = \Pr[MCC|\overline{MAN}] \cdot \Pr[\overline{MAN}] + \Pr[MCC|MAN] \cdot \Pr[MAN]. \quad (2)$$

The meanings of MCC and MAN are in Table I, so are all the symbols. We can solve (2) via (3), (4) and (9). In M-cube, the only reason why an MC is created is that v misses one ANC packet from a neighbor (*i.e.*, a MAN happens). So,

$$\Pr[MCC|\overline{MAN}] = 0. \quad (3)$$

Let i be any neighbor node of v . Therefore, $\Pr[MAN]$ is equal to $\Pr[MAN_i]$. Let j be a neighbor node with which i communicates. Therefore, we have

$$\begin{aligned} \Pr[MCC|MAN] &= \Pr[MCC|MAN_i | j \in N_{iv}] \cdot \Pr[j \in N_{iv}] \\ &+ \Pr[MCC|MAN_i | j \in N_{i \setminus v}] \cdot \Pr[j \in N_{i \setminus v}]. \end{aligned} \quad (4)$$

We can solve (4) via (5), (6) and (7).

1) $\Pr[j \in N_{iv}]$ and $\Pr[j \in N_{i \setminus v}]$

Assume that i uniformly communicates with one of its neighbors, so $\Pr[j \in N_{iv}]$ is approximately equal to the ratio

between the average intersection area of two circles (centered at v and i , respectively) and the area of an arbitrary circle. The circle represents the communication area of a node, and the radius of a circle (denoted as r) represents the communication range of a node, respectively. Assume that the communication range of a node is equal to the interference range of a node, and all nodes are deployed in an area according to a two-dimensional Poisson point process. It can be derived from [19] that the average intersection area of two circles centered at v and i is $1.84r^2$, approximately. So,

$$\Pr[j \in N_{iv}] \approx 1.84/3.14, \quad \Pr[j \in N_{i \setminus v}] \approx 1.30/3.14. \quad (5)$$

2) $\Pr[MCC|MAN_i|j \in N_{i \setminus v}]$

If j is not a neighbor of v (i.e., $j \in N_{i \setminus v}$) and v misses the ANC from i (i.e., a MAN_i happens), an MCC will definitely happen, because v misses its only chance to obtain the information that a DC is occupied by i and j . Therefore,

$$\Pr[MCC|MAN_i|j \in N_{i \setminus v}] = 1. \quad (6)$$

3) $\Pr[MCC|MAN_i|j \in N_{iv}]$

If $j \in N_{iv}$ and v misses the ANC from i (i.e., a MAN_i happens), then a MCC happens if and only if MAN_j happens, because v has another chance to overhear j 's ANC . So,

$$\Pr[MCC|MAN_i|j \in N_{iv}] = \Pr[MAN_j|MAN_i|j \in N_{iv}]. \quad (7)$$

$\Pr[MAN_j|MAN_i|j \in N_{iv}]$ is equal to the probability that v misses two ANC s from i and j , so it can be derived from [19]

$$\Pr[MAN_j|MAN_i|j \in N_{iv}] < q + (1 - q) \cdot (T_{DC} - T_{AN})/T_{DC}. \quad (8)$$

B. Derivation of $\Pr[MAN_i]$

We can solve $\Pr[MAN_i]$ via the probability that the complementary event of MAN_i happens, therefore

$$\Pr[MAN_i] = 1 - \Pr[\overline{MAN}_i]. \quad (9)$$

Moreover, by TPT, we have

$$\Pr[\overline{MAN}_i] = \Pr[\overline{MAN}_i|IIS] \cdot \Pr[IIS] + \Pr[\overline{MAN}_i|\overline{IIS}] \cdot \Pr[\overline{IIS}]. \quad (10)$$

We can solve (10) via (11), (12) and (13).

1) $\Pr[IIS]$ and $\Pr[\overline{IIS}]$

Based on the assumption we make in subsection A, i is an arbitrary neighbor of v . Therefore, in the long run, we have

$$\Pr[IIS] \approx \Pr[\overline{IIS}] \approx 1/2. \quad (11)$$

2) $\Pr[\overline{MAN}_i|IIS]$ and $\Pr[\overline{MAN}_i|\overline{IIS}]$

In M-cube, if i is a sender (i.e., an IIS happens), a \overline{MAN}_i happens if and only if three following conditions are satisfied. (1) v is on the CC at the time (denoted as $t_{i,A}$) that i starts to send an ANC (i.e., an $OCC_v(t_{i,A})$ happens). (2) v is not sending in the interval (denoted as $T_{i,A}$, i.e., $(t_{i,A}, t_{i,A} + T_{AN})$) that i is sending an ANC (i.e., a $NSC_v(T_{i,A})$ happens). (3) All the neighbors of v except i do not interfere with v on the CC in the interval $T_{i,A}$ (i.e., $\cap_{u \in N_v \setminus \{i\}} NIC_u(T_{i,A})$ happens). So,

$$\Pr[\overline{MAN}_i|IIS] = \Pr[OCC_v(t_{i,A}), NSC_v(T_{i,A}), \cap_{u \in N_v \setminus \{i\}} NIC_u(T_{i,A})]. \quad (12)$$

Similarly, if i is a receiver, we have

$$\Pr[\overline{MAN}_i|\overline{IIS}] = \Pr[OCC_v(t'_{i,A}), NSC_v(T'_{i,A}), \cap_{u \in N_v \setminus \{i\}} NIC_u(T'_{i,A})]. \quad (13)$$

Due to space limitation and the similarity between (12) and (13), we just show that how to solve (12) via (14).

In M-cube, if v is on the CC at $t_{i,A}$ (i.e., an $OCC_v(t_{i,A})$ happens), a $NSC_v(T_{i,A})$ will always happen since M-cube does not allow v to send when its neighbor i is sending an ANC . This may cause packet collisions. So if we assume $OCC_v(t_{i,A})$ and $NIC_u(T_{i,A})$ to be independent, we have

$$\Pr[\overline{MAN}_i|IIS] \approx \Pr[OCC_v(t_{i,A})] \cdot \Pr[\cap_{u \in N_v \setminus \{i\}} NIC_u(T_{i,A})]. \quad (14)$$

We can solve (14) via (15) and (17).

a) $\Pr[OCC_v(t_{i,A})]$

For v , $t_{i,A}$ is an arbitrary time. Therefore, we have

$$\Pr[OCC_v(t_{i,A})] = p_{cc}. \quad (15)$$

Let T_0 be a sufficiently long time. In T_0 , the total number of arrival messages at each node is equal to $\lambda T_0 / AVG$, so the total time that a node sends all these messages on DCs is equal to $\lambda T_0 T_{DC} / AVG$. Whereas, approximately, the total time that a node receives all these messages is $p_{rcv}(1 - p_{cc} - p_{slp})T_0$. In the long run, the total time that a node sends messages is equal to the total time that a node receives messages, when the networks are stable. In the long run, if we assume $p_{rcv} \approx 1/2$, based on $p_{cc}/p_{slp} > q$, we have

$$p_{cc} > (1 - 2\lambda T_{DC} / AVG) / (1 + 1/q). \quad (16)$$

b) $\Pr[NIC_u(T_{i,A})]$

If $u \in N_{vi}$, then a $NIC_u(T_{i,A})$ always happens because whichever channel u is on at $t_{i,A}$, M-cube does not allow u to send on the CC while its neighbor i is sending ANC , because this may cause packet collisions. Therefore, based on (5),

$$\Pr[\cap_{u \in N_v \setminus \{i\}} NIC_u(T_{i,A})] = \Pr[NIC_u(T_{i,A})]^{N_{v \setminus i}} = \Pr[NIC_u(T_{i,A})]^{1.3N}. \quad (17)$$

By the TPT, we have

$$\Pr[NIC_u(T_{i,A})] = \Pr[NIC_u(T_{i,A})|u \in N_{vi}] \cdot \Pr[u \in N_{vi}] + \Pr[NIC_u(T_{i,A})|u \in N_{v \setminus i}] \cdot \Pr[u \in N_{v \setminus i}]. \quad (18)$$

We can solve $\Pr[u \in N_{vi}]$ and $\Pr[u \in N_{v \setminus i}]$ by the same method in (5), so we can solve (18) via (19) and (20).

c) $\Pr[NIC_u(T_{i,A})]$

If $u \in N_{vi}$, based on the same reason as (17), we have

$$\Pr[NIC_u(T_{i,A})|u \in N_{vi}] = 1. \quad (19)$$

By the TPT, we have

$$\Pr[NIC_u(T_{i,A})|u \in N_{v \setminus i}] = \Pr[NIC_u(T_{i,A})|u \in N_{v \setminus i}][SLP_u(T_{i,A})] \cdot \Pr[SLP_u(T_{i,A})] + \Pr[NIC_u(T_{i,A})|u \in N_{v \setminus i}][SLP_u(T_{i,A})] \cdot \Pr[SLP_u(T_{i,A})]. \quad (20)$$

We can solve (20) by (21), (22) and (23).

d) $\Pr [NIC_u(T_{iA})|u \in N_{v \setminus i}]$

In a long run, it can be derived from [19] that

$$\Pr [SLP_u(T_{iA})] = \frac{T_{SP} - |T_{iA}|}{T_{WK} + T_{SP}} = \frac{T_{SP} - T_{AN}}{T_{WK} + T_{SP}}. \quad (21)$$

If a $SLP_u(t_{iA})$ happens, a $NIC_u(T_{iA})$ always happens because a sleeping node does not interfere with any node. So,

$$\Pr [NIC_u(T_{iA})|u \in N_{v \setminus i}|SLP_u(T_{iA})] = 1. \quad (22)$$

If u is not sleeping during T_{iA} , then by TPT, we have

$$\begin{aligned} &\Pr [NIC_u(T_{iA})|u \in N_{v \setminus i}] = \\ &\Pr [NIC_u(T_{iA})|u \in N_{v \setminus i}|OCC_u(t_{iA})] \cdot \Pr [OCC_u(t_{iA})] + \\ &\Pr [NIC_u(T_{iA})|u \in N_{v \setminus i}|OCC_u(t_{iA})] \cdot \Pr [OCC_u(t_{iA})]. \end{aligned} \quad (23)$$

We have solved $\Pr [OCC_u(t_{iA})]$ in (15). Therefore, we can solve (23) via (24) and (25).

e) $\Pr [NIC_u(T_{iA})|u \in N_{v \setminus i}]$

If $u \in N_{v \setminus i}$, u will not interfere with v overhearing the ANC from i if and only if u keeps silent during $(t_{iA} - T_{AN}, t_{iA} + T_{AN})$, i.e., no packet arrived at MAC layer of u . Therefore, according to the Poisson arrival process, we have

$$\Pr [NIC_u(T_{iA})|OCC_u(t_{iA})] = e^{-2\lambda T_{AN}}. \quad (24)$$

If $u \in N_{v \setminus i}$, by the TPT, we have

$$\begin{aligned} &\Pr [NIC_u(T_{iA})|\overline{OCC_u(t_{iA})}] = \\ &\Pr [NIC_u(T_{iA})|\overline{OCC_u(t_{iA})}|BTC(T_{iA})] \cdot \Pr [BTC(T_{iA})] + \\ &\Pr [NIC_u(T_{iA})|\overline{OCC_u(t_{iA})}|BTC(T_{iA})] \cdot \Pr [BTC(T_{iA})]. \end{aligned} \quad (25)$$

We can solve (25) via (26), (27) and (28).

f) $\Pr [NIC_u(T_{iA})|\overline{OCC_u(t_{iA})}]$

Since when u switches to the DC is unknown, the time when u switches its current channel back to the CC is uniformly distributed in the interval T_{iA} . Therefore, we have

$$\Pr [BTC(T_{iA})] = |T_{iA}|/T_{DC} = T_{AN}/T_{DC}. \quad (26)$$

Since in $(t_{iA}, t_{iA} + T_{DC})$ u does not switch back to the CC, u does not interfere with any node on the CC. So,

$$\Pr [NIC_u(T_{iA})|\overline{OCC_u(t_{iA})}|BTC(T_{iA})] = 1. \quad (27)$$

Let Δt be the duration that u is on the CC after u switches back to the CC. Based on the similar reason as (26), Δt is uniformly distributed in the interval $(0, T_{AN})$. Therefore, by the expectation of random variable function, we have

$$\begin{aligned} \Pr [NIC_u(T_{iA})|\overline{OCC_u(t_{iA})}|BTC(T_{iA})] &= E[e^{-\lambda \Delta t}] \\ &= \frac{1 - e^{-\lambda T_{AN}}}{\lambda T_{AN}}. \end{aligned} \quad (28)$$

V. PERFORMANCE EVALUATION

In this section, we performed both simulation and real testbed experiments to evaluate the performance of M-cube.

A. Simulation Results

We implemented a simulator using C++, which has 289 nodes whose transmission ranges are set to 40m. The nodes are uniformly deployed in a square area of size $200\text{m} \times 200\text{m}$ with a node density of 38 (i.e., a node that is not in the edge of networks has 37 neighbors). The many-to-many transmission model is used where the payload size is set to 32 Bytes and the channel bandwidth is set to 250 Kbps.

To investigate the effect of multiple channel reservation and dynamic duty cycling, M-cube is compared with another four famous schemes: (1) CSMA\CA; (2) MMSN [1]; (3) PMC [3]; (4) CAM-MAC [4]. Two varieties of M-cube are also implemented for comparisons. The first one utilizes Single-Channel Reservation, called SCR, which is used to justify the effect of multiple channel reservation. The second one exploits a Fixed Duty Cycle of 50%, called FDC, which is used to justify the effect of dynamic optimal duty cycling. Three groups of simulations were conducted to examine four metrics: throughput, packet delivery ratio, and energy consumption. In each group, different Total Number of Channels (TNC) and the loads are considered. TNC includes the CC and all DCs, and the loads are varied via changes of the Number of CBR (NCBR, Constant Bit Rate) streams in the networks. In all the simulations, TNC is set to 4 while NCBR varies; NCBR is set to 30 for different TNCs.

1) *Evaluation of throughput*: The throughput is computed as the total number of all the useful data successfully delivered per unit time.

The effect of TNC on throughput is shown in Fig.3 (a). M-cube has lower throughput when TNC is smaller than 3. Besides duty cycling, this is also due to that under multiple channel reservation of M-cube all node-pairs have to switch back to the CC first to send an ANC, and then communicate on a DC. This scheme will pay a considerable cost if TNC is small. When more channels are available, M-cube, CAM-MAC and PMC allow more nodes to communicate on different DCs simultaneously. This is because they employ dynamic channel selections, and thus outperform CSMA and MMSN. However, when TNC becomes larger than 5, M-cube performs a little better than CAM-MAC and PMC. This is because CAM-MAC suffers from collisions of cooperative packets and PMC suffers from THT, whereas M-cube avoids using cooperative packets and tackles THT by multiple channel reservation, so it achieves higher throughput.

The effect of loads on throughput is shown in Fig.3 (b). It is observed that the throughputs of all the protocols rise with NCBR. This is because if more node-pairs are involved in communications, more simultaneous transmissions will occur on the DCs. With light loads, M-cube underperforms the others. However, the results show that with heavy loads, M-cube performs progressively better than the other protocols, which indicates that M-cube significantly benefits from the multiple channel reservation when the degree of THT increases with the loads, even though it is still duty cycling. Note that M-cube outperforms FDC and SCR when NCBR is larger than 32.

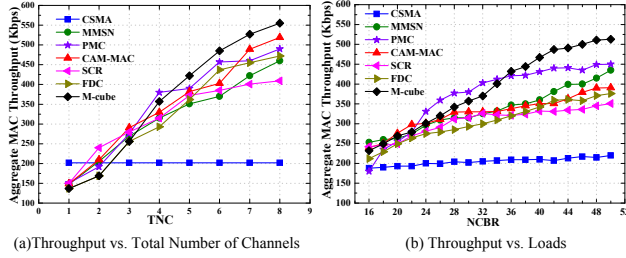


Figure 3. Throughput evaluation

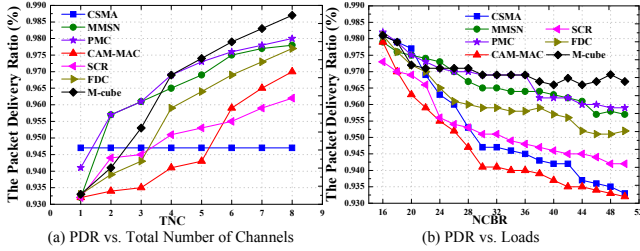


Figure 4. Packet delivery ratio evaluation

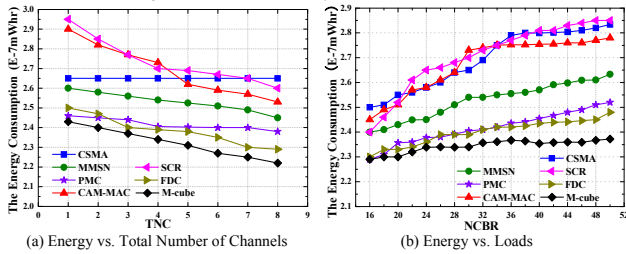


Figure 5. Energy consumption evaluation

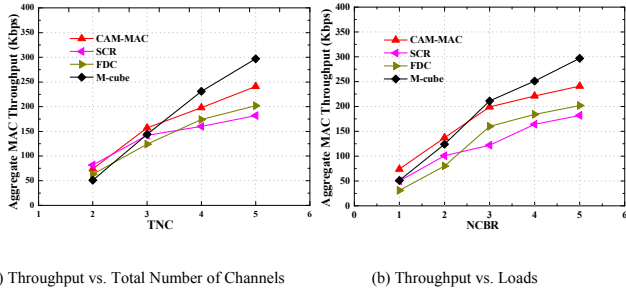


Figure 6. Testbed evaluation on throughput

2) *Evaluation of packet delivery ratio*: Packet delivery ratio (PDR) is computed as the ratio of the total number of packets successfully delivered over the total number of packets requested to be delivered.

The effect of TNC on PDR is shown in Fig.4 (a). The results show that all PDRs increase with the rise of TNC. When TNC is smaller than 4, MMSN and PMC achieve better performances than M-cubes and CAM-MAC. One possible reason is that the schemes of CAM-MAC and M-cubes for tackling THT undermine PDR. However, when TNC is larger than 5, M-cube performs better than the others, but FDC and SCR still perform worse than MMSN and PMC. This is primarily because M-cube does not involve a retransmission scheme. In addition, FDC has a fixed duty cycle and senders under SCR drop some packets because that the single channel reservation cannot reserve a DC in time.

The effect of loads on PDR is shown in Fig.4 (b). It is observed that all PDRs generally drop when the loads are heavier except that of M-cube, which keeps stable around 96%-97%. This is because that under multiple channel reservation, node-pairs are more likely to find an idle DC for communication timely before the packets are dropped by the sender due to packet lifetime expiration. FDC outperforms SCR, which verifies benefit of multiple channel reservation. M-cube outperforms FDC and SCF, which is still caused by fixed duty cycle and flaws of single channel reservation.

3) *Evaluation of Energy Consumption*: The consumption of energy for all the schemes is computed as the consumed energy to successfully deliver a useful data byte.

The effect of TNC on energy consumption is shown in Fig.5 (a). The results show that the energy consumptions of all the protocols decrease with the rise of TNC, but M-cube always outperforms the others. This means that M-cube conserves more energy to prolong network lifetime without employing time synchronization of MMSN and continuous channel switching of PMC. Moreover, CAM-MAC always consumes more energy than the others due to cooperative packet collisions, which undermines many communications. Finally, note that when TNC becomes larger, the gap between M-cube and FDC on energy consumption becomes larger, which indicates that M-cube with dynamic duty cycle is capable of achieving higher energy efficiency for the networks with more DCs.

The effect of loads on energy consumption is shown in Fig.5 (b). All energy consumptions increase when loads rise. M-cube maintains lower energy consumption when NCBR is larger than 25. This is because the other protocols suffer from certain problems. MMSN consumes much energy to maintain time synchronization among all the nodes when loads are heavy; PMC has many collisions on the current channel when loads are heavy; CAM-MAC suffers from the collisions between cooperative packets and reservation packets when more nodes communicate simultaneously. FDC and SCR suffer from fixed duty cycling and single channel reservation.

B. Testbed Experiment Results

We built a sensor node platform, Hawk, for our experiments. Several experiments were conducted to evaluate M-cube's performance. Hawk employs $\mu\text{C}/\text{OS}$, where each node is equipped with an nRF905 radio and a MSP430 processor. A hawk node is shown in Fig.7 (a). The testbed consists of 10 hawk nodes which are completely connected as shown in Fig.7 (b). 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 communication. The experiment was repeated for 10 times. When an experiment is finished, all the nodes send their total number of bytes received during the 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 M-cube, SCR, FDC and CAM-MAC were implemented for throughput comparisons.

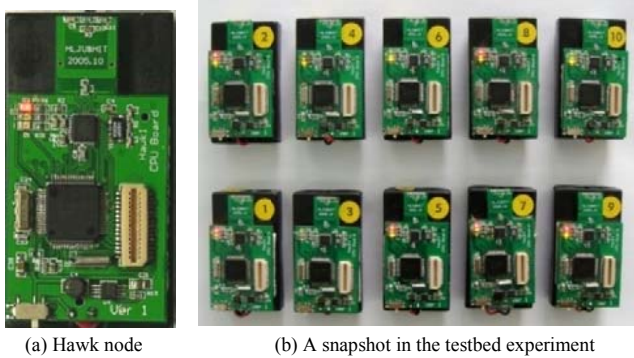


Figure 7. Testbed experiments

The effect of TNC on throughput is shown in Fig.6 (a), with NCBR set to 5. It is observed that CAM-MAC has higher throughput than M-cube when TNC is less than 4. This is primarily because CAM-MAC does not have to enable duty cycling or switch among DCs, which undermine the throughput of M-cube when THT is less severe. Nevertheless, M-cube achieves better throughput when TNC is larger than 4. This is because as more DCs are available, THT becomes more serious, and M-cube tackles THT with less cost than CAM-MAC. These results are almost consistent with simulation results shown in Fig.3 (a). M-cube has similar throughput with SCR and FDC when THT is small; whereas M-cube outperforms them when THT is larger than 3.

The effect of loads on throughput is shown in Fig.6 (b), with NCBR set to 5. It shows that three M-cube based protocols have lower throughput than CAM-MAC when loads are small. This is because that when fewer nodes are involved in communication, the cooperative scheme of CAM-MAC works better to tackle THT than multiple channel reservation of M-cube. However, when the loads are becoming heavier, fewer nodes are left as cooperative neighbors to send cooperative packets, which are employed by CAM-MAC to prevent THT. Therefore, M-cube outperforms CAM-MAC when NCBR is equal to or larger than 3. This is also because that when the loads are heavier, M-cube avoids the collision between cooperative packets and reservation packets on the CC under the cooperation scheme in CAM-MAC. Finally, note that M-cube works better than SCR and FDC.

VI. CONCLUSION

The triple hidden terminal problems are major reasons of energy wastage in WSNs. To address these problems, a dynamic duty cycling based MAC protocol, M-cube, with multiple channel reservation is proposed. Extensive simulations were conducted to examine the performance of M-cube. The results show that with multiple channel reservation, M-cube can solve the triple hidden terminal problems with a lower cost, and still enable duty cycling at the same time. Thereby, M-cube achieves a significant improvement of the energy efficiency and other performances as well, especially when the total number of channels and loads increase. We also implemented M-cube on a real sensor platform. The testbed results show that both dynamic optimal duty cycling scheme and multiple channel reservation actually enable M-cube to achieve better throughput.

VII. 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 (2008RFQXG107), the Heilongjiang Postdoctoral Science Foundation (No.LRB08-021), the Science and Technology Key Research of Heilongjiang Educational Committee (1154Z1001), the Heilongjiang University Foundation for the Outstanding Young Scholar, the Student Innovative Research Project (2010208) and Innovative Laboratory Project (2010046).

REFERENCES

- [1] G. Zhou, C. Huang, T. Yan, *et al.* MMSN: Multi-Frequency Media Access Control for Wireless Sensor Networks, in *INFOCOM*, 2006.
- [2] Y. Kim, H. Shin, and H. Cha. Y-MAC: An Energy-efficient Multi-Channel MAC Protocol for Dense Wireless Sensor Networks, in *IPSN*, 2008.
- [3] H. K. Le, D. Henriksson, *et al.* A Practical Multi-Channel Media Access Control Protocol for Wireless Sensor Networks, in *IPSN*, 2008.
- [4] T. Luo, M. Motani, *et al.* Cooperative Asynchronous Multichannel MAC: Design, Analysis, and Implementation, *IEEE TMC*, Vol 8, No 3, 2009.
- [5] M. Salajegheh, *et al.* HyMAC: Hybrid TDMA/FDMA Medium Access Control Protocol for Wireless Sensor Networks, in *PIMRC*, 2007.
- [6] M. Jovanovic, and G. Djordjevic. TFMAC: Multi-Channel MAC Protocol for Wireless Sensor Networks, in *TELSIKS*, 2007.
- [7] G. p. Halke and K.G. Langendoen. Crankshaft: An energy-efficient MAC protocol for dense wireless sensor networks, in *EWSN*, 2007.
- [8] Y. Wu, John A. Stankovic, *et al.* Realistic and Efficient Multi-Channel Communications in Wireless Sensor Networks, in *INFOCOM*, 2008.
- [9] J. Ansari, T. Ang, and P. Mahonen. Spectrum Agile Medium Access Control Protocol for Wireless Sensor Networks, in *SIGCOMM*, 2009.
- [10] 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.
- [11] A. Adya, P. Bahl, J. Padhye, and A. Wolman. A Multi-Radio Unification Protocol for IEEE 802.11 Wireless Networks, in *BROADNETS*, 2004.
- [12] 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.
- [13] A. Nasipuri, J. Zhuang, and S. R. Das. A Multichannel CSMA MAC Protocol for Multihop Wireless Networks. in *WCNC*, 1999.
- [14] J. So and N. Vaidya. Multi-Channel MAC for Ad Hoc Networks: Handling Multi-Channel Hidden Terminals Using a Single Transceiver, in *Mobihoc*, 2004.
- [15] J. Chen, S. Sheu, and C. Yang. A New Multichannel Access Protocol for IEEE 802.11 Ad Hoc Wireless LANs, in *PIMRC*, 2003.
- [16] A. Tzamaloukas and J.Garcia-Luna-Aceves. Channel-Hopping Multiple Access with Packet Trains for Ad Hoc Networks, in *MoMuC*, 2000.
- [17] 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.
- [18] T. Luo, M. Motani, and V. Srinivasan. Altruistic Cooperation for Energy-Efficient Multi-Channel MAC protocols, in *Mobicom*, 2007.
- [19] T. Luo, M. Motani, *et al.* Analyzing DISH for Multi-Channel MAC Protocols in Wireless Networks, in *Mobihoc*, 2008.