CT-MAC: Energy-Efficient Contention-based MAC Protocol for Wireless Sensor Networks

Milica D. Jovanovic[1] and Goran Lj. Djordjevic[2]

***Abstract* – This paper presents Control-Tone MAC (CT-MAC), a scheduled contention-based medium access protocol especially designed for Wireless Sensor Networks (WSNs). Similarly to other contention-based MAC protocols with common active periods (e.g. S-MAC, T-MAC, and SCP-MAC), CT-MAC coordinates sensor nodes into sleep/wakeup schedules, allowing them to remain awake only for brief contention periods. Unlike most of the current solutions, CT-MAC employs short control tones, instead of control packets (e.g. RTS/CTS) in order to realize an energy-efficient contention resolution mechanism in multi-hop networks. The simulation results demonstrate that CT-MAC significantly reduces energy waste due to collisions, overhearing and idle listening in respect to SCP-MAC.**

***Keywords* – wireless sensor networks, medium access control, energy efficiency, contention protocol.**

I. Introduction

Recent advances in wireless communications, low-power design, and MEMS-based sensor technology have enabled the development of relatively inexpensive and low power wireless sensor nodes. The common vision is to create a large wireless sensor network (WSN) through *ad-hoc* deployment of hundreds or thousands of such tiny devices able to sense the environment, compute simple task and communicate with each other in order to achieve some common objective, like environmental monitoring, target tracking, detecting hazardous chemicals and forest fires, monitoring seismic activity, military surveillance [1]. The primary objective in WSN design is maximizing node/network lifetime, leaving the other performance metrics as secondary objectives. Since the communication of sensor nodes is more energy consuming than their computation, it is a primary concern to minimize communication while achieving the desired network operation.

A medium access control (MAC) protocol decides when competing nodes may access the radio channel, and tries to ensure that no two nodes are interfering with each other’s transmissions. MAC protocols for WSN usually trade off performance (latency, throughput, fairness) for cost (energy efficiency, reduced algorithmic complexity), while providing a good scalability and some limited adaptability for topology changes [2]. Collisions, overhearing, and idle listening are the main types of energy waste for sensor nodes that occur during medium access. Collision occurs if a node receives multiple transmissions at the same time. Idle listening occurs if a node listens to the medium when there is no transmission, whereas an overhearing happens when a node receives a data packet transmission even if it is not the intended recipient of this transmission.

One important approach is based on common active/sleep periods. Nodes use active periods for communication and the sleep periods for saving energy. At the beginning of each active period, nodes contend for the medium using contention-based approaches. Only nodes participating in data transfer remain awake after contention periods, while others can sleep. Contention-based media access mechanisms of various kinds are employed in a plurality of different MAC protocol for WSNs. For instance, S-MAC employs an explicit contention mechanism which requires carrier sense and the use of control packets, such as Request to Send (RTS) and Clear to Send (CTS) [3]. Although RTS/CTS can alleviate the hidden terminal problem, it incurs high overhead because data packets are typically very small in WSN [4]. SCP-MAC replaces RTS/CTS control packets with short wake up tones [5]. When a node wakes up during the common active schedule and does not find a tone, it goes back to sleep. In order to improve the contention performances, SCP-MAC introduce a two level contention window. Before sending the tone, a node performs carrier sense by randomly select a slot within the first contention window. If the channel is idle, then the node sends the tone to wake up the receiver. Only nodes that successfully send wakeup tones will enter the second contention window. Such nodes randomly select a slot in the second contention window and then perform a carrier sense; if they find the channel idle then they transmit the data. The major advantage of spitted contention phase is lower collision probability with shorter overall contention time. Although the SCP-MAC is more energy efficient than S-MAC, it does not implement an appropriate mechanism to alleviate the hidden terminal problem that normally exists in multi-hop networks.

In this paper, we present CT-MAC, abbreviated from Control Tone MAC, which uses short signal tone transmissions to implement an energy-efficient contention mechanism. In CT-MAC we use control tones not only to wakeup intended receivers but to implement efficient handshake mechanism among competing nodes arranged in a multi-hop network.

II. The Protocol Description

In our contention protocol, competition is accomplished by exchanging short *control tones* among competing nodes. Control tones play a similar role as RTS/CTS packets in traditional collision avoidance handshaking mechanism with

[1, 2] Milica D. Jovanovic and Goran Lj. Djordjevic are with Faculty of Electronic Engineering, University of Nis, Aleksandra Medvedeva 14, P.O. Box 73, 18000 Nis, Serbia, E-mails: {milica.jovanovic, gdjordj}@elfak.ni.ac.rs

the following two important differences: (a) control tone is not a packet; it is a simple flag signal and encodes no data, and (b) control tones may collide with one another without affecting their functionality. Control tones are transmitted over a shared wireless medium and nodes within the transmission range individually decide how to react to those tones, based on their own states at that time. Thus, different from a handshake in other contention protocols, the handshake in CT-MAC is implicit in a sense that a node that hears a control tone does not know the identity of the sending node neither it knows whether it hears one tone or a mix of several control tones.

The contention period is slotted. In each slot, a node can either transmits the control tone, listen for a tone, or stay inactive. The duration of the slot is equal $T_s = T_d + T_g$, where T_d is the duration of the control tone, and T_g is the guard time which compensate time discrepancy among nodes. Note that the control tone can be very short in duration, just enough for the listening node to check if the medium is busy (typically 2 – 3 ms). The guard time depends on clock drift rate and the efficiency of synchronization mechanism employed. Typical value of the guard time is in order of several milliseconds.

During contention, a node can be put into one of the following five states: P_SEND, P_REC, IDLE, SEND, and REC. A node enters contention period either in state P_SEND or P_REC. The P_SEND is for potential sender, i.e. node which has data packet to send in the current frame, and the P_REC is for potential receiver, i.e. node which does not have data to send. Node leaves the contention period in one of three final states, IDLE, SEND or, REC, which controls its behavior during data transmission phase. A node in IDLE state will be put into sleep mode until the next frame; a node in state SEND is a potential sender which is chosen to send its data packet, and node in state REC is a node that is instructed to enter data transmission phase in the listening mode in order to receive a data packet intended to it (if any).

Slots of the contention period are arranged into two contention windows, CW1 and CW2, with N slots in CW1, and $2M$ slots in CW2, where N and M are network-wide constants. Nodes that survive the contention window CW1 enter the CW2. Nodes compete for the medium by following a different contention procedure in each contention window. In what follows, we provide a detailed description of these two contention procedures.

Contention procedure for CW1. CT-MAC requires that each node is assigned a listening slot in contention window CW1. Each node also knows the numbers of listening slots assigned to all neighbors in its transmission range. A potential sender announces its intention to send a data packet by transmitting the control tone in the listening slot of the intended receiver. On the other hand, a node treats the presence of a control tone in its listening slot as an indication that there is **at least one** adjacent node that has packet for it.

The contention procedure is illustrated by flow diagram in Fig 1. A node enters this procedure only for its own listening slot and listening slots of neighboring nodes. A potential receiver, say node R, skips all slots (i.e. stays inactive) until its listening slot (denoted with *MySlotID* in Fig. 1). If a control tone is not heard during this slot, node will modify its state to IDLE. Otherwise, if a control tone is heard, the node R will change its state to REC. Once put in state REC, the node will start to monitor the channel during listening slots of neighboring nodes. If a control tone is heard in any of monitored slots, note R will change its state to IDLE and quit the competition.

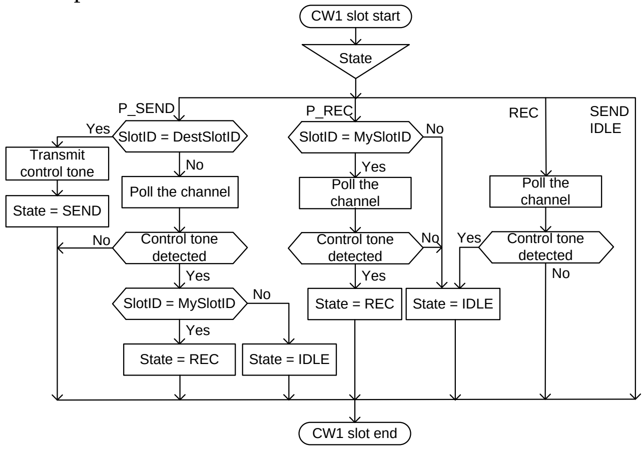


Fig. 1. Flow diagram for CW1 contention procedure.

While waiting for the listening slot of its intended receiver (denoted with *DestSlotID* in Fig. 1), a potential sender, say node S, monitors the channel for control tone transmissions during listening slots of neighboring nodes. If a control tone is heard, node S will quit the competition in IDLE state. An idle channel will allow the node S to proceed until the listening slot of its intended receiver, when it transmits the control tone and modifies its state to SEND. Once in SEND state, the node S will skip all slots until the end of CW1. It may happen that the listening slot of the node S precedes the listening slot of its intended receiver. In that case, S will poll the channel in its own listening slot. If a control tone is not heard, S will resume its operation as potential sender. Otherwise, if control tone is heard, the S will change its state to REC and continues slot monitoring, but now as a potential receiver.

The contention procedure for CW1 in a network with 8 nodes is illustrated in Fig 2(a) represents an initial condition. Potential senders are represented with dark gray circles, and potential receivers are represented with light gray circles. Every two nodes within the transmission range of one another are connected with an edge. Arrow on the edge points to the intended receiver. The numbers enclosed with brackets attached to each node denote the number of the listening slot assigned to that node. The first control tone transmission occurs in slot 2 (i.e. in the listening slot of node N_2). In this slot, node N_1 transmits the control tone to notify node N_2, along with all potential senders within the range, of its intent to send a data packet to N_2. As a consequence, N_1 changes its state to SEND. Presence of a control tone in its own listening slot moves N_2 into REC state. In contrast to N_1, which will stay inactive until the end of CW1, N_2 continues to monitor listening slots of neighboring nodes. Slot 2 is also monitored by N_2's surrounding potential senders, N_3 and N_4. Since N_3 hears the tone it changes its state to IDLE - giving up its data transmission in order to prevent a possible collision at node N_2. On the other hand, node N_4 does not hear anything and continues its operation as potential sender. The next control tone is transmitted by node N_7 during the slot 3. This transmission affects only nodes N_7 and N_8 since there is no other node that hears the tone. Note that N_5 is potential receiver which will stay inactive until its own listening slot. The third and the final control tone transmission takes place in slot 4. Now we have two potential senders, N_4 and N_6, transmitting control tone in the same slot. In spite of tone collision, N_5 detects channel activity and changes its state to REC. Nodes N_4 and N_6 are not aware of one another transmission, and both set its state to SEND. Given that potential senders N_3 and N_7 have already determined their final state in CW1, the node N_2 is the only N_5's neighbor that monitors its listening slot. Since N_2 hears the tone transmitted by N_4, it becomes aware that there are now two senders in its neighborhood and that the collision of their data packet is inevitable. This is the reason why it changes its state from REC to IDLE. Until the end of CW1 there are no control tone transmissions, and no state changes. The node state distribution after CW1 can be seen in Fig 2(b). Note that if data transmission starts immediately after CW1, there will be one successful data transfer (from N_7 to N_8) in addition to one collision (at node N_5) and one unnecessary data packet transmission (N_1).

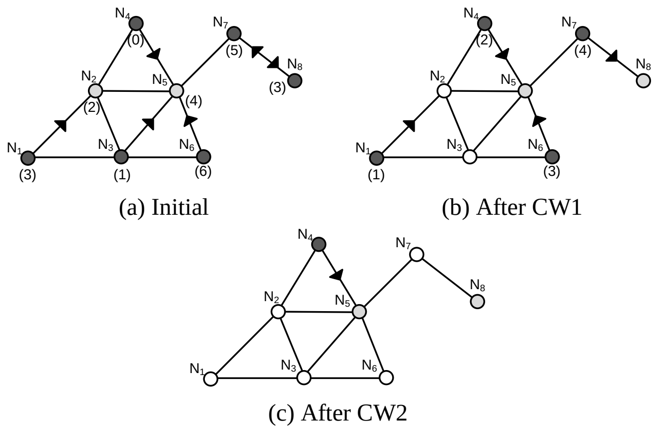


Fig. 2. Distribution of sending, receiving and idle nodes during contention period of CT-MAC – an example.

Contention procedure for CW2. Although contention in CW1 eliminates a number of conflicting conditions, there are still situations that may cause energy loss during data transfer. This is the reason why we introduce the second contention window, CW2. Active participants in this round of contention are nodes that survive competition in CW1, i.e. those in SEND and REC state. CW2 is divided in two equal sized sub-windows, CW2a and CW2b. Slots in both sub-windows are numbered form 0 to M-1. The task of a sending node in CW2a is to randomly pick a slot number from the range [0, M-1], remember that number for future use, and then transmit control tone in the selected slot. The operation of a receiving node is a little bit involved. A receiving node wakes up at the beginning of CW2a and continues to listen in each slot until it hears a control tone. It remembers the number of the corresponding slot, and then skips remaining slots in CW2a.

During sub-window CW2b, competing nodes change their roles - nodes in REC state are those who will transmit, and nodes in state SEND are those who will listen for control tones. The only responsibility of a receiving node in CW2b is to transmit a control tone in slot *i*, where *i* is the number of slot in CW2a during which it was heard the tone. On the other hand, a sending node wakes up at the beginning of CW2a and continues to listen in each slot until it hears a control tone. If a control tone is heard in the slot with the same number as one that it was used to transmit a control tone in CW2a, the sending node will keep SEND state skipping the remaining slots in CW2b. Otherwise, if a control tone is heard in some other slot, the sending node will quit contention in IDLE state.

The contention procedure for CW2 is primarily designed to eliminate collisions that occur when multiple senders transmit data packets to their common neighbor. This situation is illustrated in Fig. 2(b) where both nodes N_4 and N_6 are about to send data packet to node N_5. The numbers enclosed with brackets now denote the slot numbers in CW2a chosen by sending nodes. The collision will be prevented if these two nodes chose to transmit their control tones during different slots in CW2a. The winning sender is one which selects the slot with the smallest number in CW2a.

The reason why a sending node listens not only during the slot of CW2b in which it expects a response from its intended receiver, but also during all previous slots is to prevent collisions that its data packet may induce at some other receiving nodes within its transmission range. Let consider operation of node N_7 in Fig. 2(b). While waiting for a response from its intended receiver N_8, node N_7 is in slot monitoring mode. In slot 2, node N_5 transmits a control tone as a response to the tone that it was heard in the corresponding slot of CW2a. Since N_7 is within the transmission range of N_5, it will hear that tone and modify its state to IDLE. Node N_8 cannot know that N_7 is no longer in SEND state, and it will keep its REC state after transmitting confirmation control tone in slot 4. This creates a condition for idle listening. However, idle listening is less energy costly than a packet collision that would happen if N_7 was not given up its data transmission.

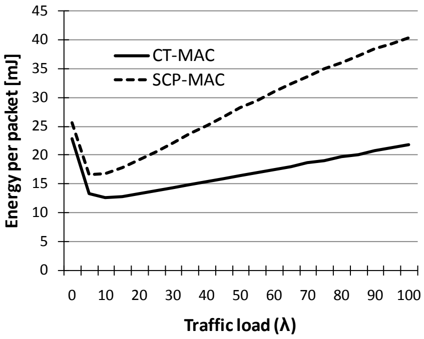
Finally, let see what happens with node N_1. Node N_1 enters CW2 in SEND state, even though its intended receiver, N_2, is already in IDLE state. Even though N_1 transmits a control tone in CW2a (during slot 1), there is no confirmation control tone in CW2b since N_2 is already quit its activity in this frame. Because of that, N_1 quits the contention and postpones its data transmission until the next frame. The final outcome of the contention resolution can be seen in Fig. 2 (c). The only survival sender-receiver pair is N_4-N_5 in addition of one receiving node, N_8, predestinated to idle listening.

III. Performance Evaluation

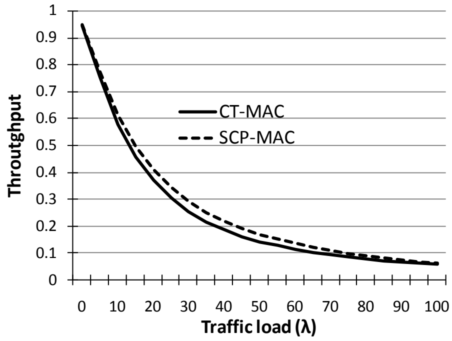
We implement CT-MAC and SCP-MAC protocols in a custom WSN simulator build in C++, and conduct several experiments to evaluate their performances. Our evaluation is based on the simulation of 200 nodes randomly distributed in an area of 100x100 m^2. The average number of nodes within radio transmission range, i.e. the average size of one-hop neighborhood, in this network is 6. Data rate of 20 Kbps is assumed, and data packet length is fixed to the value of 32 bytes. Both analyzed protocols divide contention period into two contention windows, CW1 and CW2. We configure both protocols with the 32 slots in CW1 and 16 slots in CW2. The

Parameter	Value
Time needed to poll channel once	3 ms
Time to transmit a control tone	5 ms
Time to transmit or receive a data packet	40 ms
Time to transmit/receive an ACK packet	18 ms
Time to receive MAC header	16 ms
Idle channel timeout period	8 ms

(a) Experimental parameters for CT-MAC



(b) Energy consumption per transferred packet.



(c) Throughput over varying traffic load.

Fig. 3. Results of performance evaluation.

receiver responds with an acknowledgement (ACK) packet after each successful data packet transmission. Also, the receiver performs overhearing avoidance by examining the destination address of a packet immediately after receiving its MAC header. If packet is destined to another node, it immediately stops the reception. Relevant timing parameters used in our analysis are shown in Fig 3(a).

We first compare the energy performance of CT-MAC and SCP-MAC by varying the traffic load. To vary the degree of traffic load, we vary the percentage, λ, of nodes in the network that generate data packet during each frame. During simulation we count successfully transferred data packets, and record all nodes' activities in order to estimate energy used by each node. Figure 3(b) shows the average energy consumption per transferred data packet as the traffic load increases.

There are two types of energy waste, energy consumed during contention period, and energy consumed during data transfer period. The contention energy is present even when data traffic is absent since each node must poll the channel once during each contention period. Under the light traffic load, almost all generated data packet are successfully transferred with small increase of node activity during contention period. As a result, the energy per packet decreases with the increase of traffic load. Further increase of traffic load results in steady increase of energy per packet. From one hand, contention energy increases since a larger number of nodes are involved in contention. From the other hand, energy consumed during data transfer period increases too, since collisions, packet overhearing and idle listening happen more often. It is clear from Fig. 3(b) that, with a more aggressive network-wide suppression of collisions and packet overhearing, the CT-MAC always outperforms SCP-MAC in terms of the energy efficiency, especially when the traffic load is high. CT-MAC uses only 10% less energy than SCP-MAC to handle the traffic load less than λ=5%. As traffic load increases, the energy saving is greater. For example, when λ is 30%, CT-MAC consumes 35% less energy than SCP-MAC per transferred data packet. Under the maximum traffic load, CT-MAC uses only about half the energy of SCP-MAC.

In the second experiment, the results of which are shown Fig. 3(c), we study the influence of the traffic load on data throughput. We define the throughput as the ratio between the number of successfully transferred packets and the number of packets that are generated per frame. Note that, even though the throughput falls, the throughput that is achieved by CT-MAC is constantly higher than in SCP-MAC case. This is due to the fact that SCP-MAC ignores the effect of hidden terminals, which limits its efficiency in a multi-hop scenario. On the other hand, CT-MAC utilizes more elaborate contention mechanism which prevents most of collisions that might be caused by hidden terminals. This not only saves the energy but also increases data throughput.

IV. Conclusion

In this paper, we describe energy-efficient, contention-based MAC protocol, called CT-MAC. The novelty of our approach resides in its use of short control tones instead of control packets in order to implement a contention scheme in a WSN. Although CT-MAC does not alleviate the hidden terminal problem entirely, it significantly reduces probability of collisions and lowers energy waste due to overhearing and idle listening in respect to SCP-MAC. In this paper, we do not consider the problems of node synchronization and contention slot assignment. This remains for a future work.

Acknowledgement

This work was supported by the Serbian Ministry of Science and Technological Development, project No. TR - 11020 - "Reconfigurable embedded systems".

References

[1] F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, ”Wireless sensor networks: A survey”, *Computer Networks*, vol. 38, no. 3, pp. 393–422, 2002.

[2] W. Ye, J. Heidemann, and D. Estrin, “Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks”, *IEEE/ACM Trans. Net.*, vol. 12, no. 3, pp. 493-506, June 2004.

[3] W. Ye, J. Heidemann, and D. Estrin, “An energy-efficient MAC protocol for wireless sensor networks”, In *21st Conference of the IEEE Computer and Communications Societies (INFOCOM)*, vol. 3, pp. 1567–1576, June 2002.

[4] A.Woo and D.Culler, “A transmission control scheme for media access in sensor networks”, In ACM MobiCom 2001, pp. 221-235, 2001.

[5] W. Ye, F. Silva and J. Heidemann, “Ultra-Low Duty Cycle MAC with Scheduled Channel Polling”, In Proc. of ACM SenSys, Boulder, CO, November 2006.