Design of a TDMA-based Multi-Channel MAC Protocol for Wireless Sensor Networks

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

***Abstract* – In this paper, we propose a TDMA-based multi-channel Multiple Access Control (MAC) protocol for Wireless Sensor Networks (WSNs) and study its performance. The proposed protocol builds on TFMAC, a MAC protocol proposed in our previous work, but extends it in several ways to improve self-configuration capabilities and autonomy of operation, as follows: (a) protocol operation is divided into two main phases: initialization phase, where sensor nodes first discover their neighbours, and then cooperatively establish TFMAC-like collision-free multichannel TDMA schedule without the need for any local or global master nodes, and active phase, when regular data communication is taking place, and (b) a new hierarchical TDMA scheme is introduced which eases both the node synchronization and the adaptation to the incremental topology changes during the active period. We demonstrate by simulations that the proposed multi-channel protocol achieves reasonable performances during self-configuration process, in terms of the duration of the initialization phase.**

***Keywords* – wireless sensor networks, medium access control, energy efficiency, TDMA-based protocol**

I. Introduction

A wireless sensor network (WSN) is made of a number of autonomous and inexpensive sensor nodes each of which composed of sensors, a low-power radio transceiver, small amount of memory and processing capability as well as limited battery power supply. The common vision is to create a large 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 common objective [1].

Media Access Control (MAC) is a key component to ensure the successful operation of WSNs and it has obtained intensive research attention [2]. A MAC protocol decides when competing nodes could access the shared medium to transmit their data and tries to ensure that no collisions occur. MAC protocol controls the activity of nodes' radio transceiver directly, and therefore makes a strong impact to the overall network performance and energy efficiency.

Most of the MAC protocols proposed for WSNs assume the use of simple, low-cost transceivers that can operate on a single channel (frequency), only. On the other hand, the current commercial, low-power transceivers, such as CC1100 [3], already provide the basic functions required to support multiple channels. Such transceiver cannot transmit and receive at the same time, but it can switch the operating frequency dynamically. Availability of multiple channels adds one more degree of freedom to wireless communications that can be exploited to increase the spatial reuse by providing more simultaneous transmissions than is possible in single-channel WSNs. Thus, network throughput can potentially be increased.

There exist several proposals for multi-channel usage in WSN [4]. MMSN [5] is a slotted CSMA protocol that assigns channels to the receivers. At the beginning of each beacon interval, potential senders for the same receiver contend for the medium on receiver's channel. Y-MAC [6] is based on TDMA access where timeslots are assigned to the receivers. In this protocol, potential senders contend for the medium on default channel at the beginning of the timeslot owned by the common intended receiver. If multiple packets need to be transmitted, then the senders and the receiver hop to a new channel according to a predetermined sequence. TFMAC is a TDMA-based multi-channel MAC protocol, proposed in our previous work [7], which divides each channel into time slots and assigns each transmitter one time slot on each channel. A distributed time slot/channel allocation scheme employed in TFMAC guaranties collision-free data communication among neighboring nodes.

In this paper, we propose a TDMA-based multi-channel MAC protocol for WSNs which builds on TFMAC but extends it in several ways to improve self-configuration capabilities and autonomy of operation. The protocol operation is divided into two main phases: *initialization phase*, where sensor nodes first discover their neighbors, and then cooperatively establish TFMAC-like collision-free multi-channel TDMA schedule, and *active phase*, when regular data communication is taking place. Also, we introduce a new hierarchical TDMA scheme which eases both the node synchronization and the adaptation to the incremental topology changes during the active period. The rest of the paper is organized as follows. In Section II, we describe the design of the enhanced TFMAC protocol. In Section III, we provide performance evaluations and analysis. Finally in Section III, we conclude the paper.

II. The Protocol Description

The protocol assumes a randomly deployed WSN with a single base station. The protocol operation comprises three main phases: (a) inactive phase, (b) initialization phase, and (c) active phase (Fig. 1)). After deployment, all nodes are in the inactive phase. During this period, nodes regularly sample the medium to check for activity on default channel. In this context, sampling means periodically measuring the received signal strength. The transition from inactive to initialization phase is initiated by the base station which transmits a wake-

[1,2] Milica D. Jovanovic and Goran Lj. Djordjevic are with UNIVERSITY OF NIŠ, FACULTY OF ELECTRONIC ENGINEERING, Aleksandra Medvedeva 14, P.O. Box 73 18000 Niš, Serbia, E-mails: {milica.jovanovic, gdjordj}@elfak.ni.ac.rs

up tone - an unmodulated continuous signal at frequency of the default channel of duration equal to the sampling period. Nodes that detect activity on the channel, first retransmit the wake-up tone, and then switch to the initialization phase. During the initialization phase, the network prepares for its working life by going through a self-organization process which aim is to synchronize the nodes, set up a multi-channel TDMA schedule, and configure the nodes with the correct control parameters. The working life i.e. the active phase of the protocol starts as soon as the organized structure is established. The network toplogy may change over the active phase: existing nodes can fail, and new nodes may be introduced. Once the network is too altered to function correctly, return to inactive phase may be necessary in order to re-initialize the network and rebuild the self-organization. The phase transition from initialization to active phase, as well as from active to inactive phase is initiated by the activation and deactivation command messages, respectively. These messages are broadcasted by the base station and then flooded through the network.

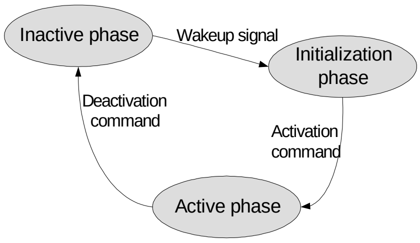


Fig. 1. The protocol phases

A. Active phase

We assume that all nodes are synchronized during the active period, and time is divided into cyclic intervals of fixed length, so called *epochs*, each having *N* super-frames. Each super-frame begins with a control slot (CS), followed by a sequence of N_F frames. CSs are used for conflict-free transmission of control packets which include information needed for the nodes to get synchronized and join the network. Each frame is divided into the *N* time slots, during which the nodes send data messages without contention and sleep when they do not have a message to send or receive. Time slot duration is suitably chosen to accommodate the transmission of one fixed-size data message. The hierarchical chart which describes the organization of time in the protocol is shown in Fig. 2. Symbols on edges in this chart indicate the number of sub-intervals contained within the parent interval. Symbols which denote the duration of the various intervals are written in brackets.

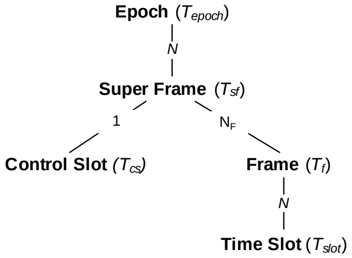


Fig. 2. The protocol time organization

The protocol requires that each node is assigned two identifiers: the pre-assigned permanent identifier (ID), which is used as a network-wide unique node's MAC address, and the local identifier (LID), which is dynamically assigned to the node during network initialization phase. LID is an integer from interval [0, *N*-1] which should be unique in node's 2-hop neighborhood. LID indicates CS in the epoch which is owned by the node. The protocol also requires each node to maintain a simple local lookup table, called *LID table*, which will associate 1-hop neighbor nodes IDs with their assigned LIDs.

The protocol provides conflict-free transmission during a particular CS slot by scheduling access among 2-hop neighboring nodes according to their LIDs. A node transmits control packet in its own CS; listens to the CSs that are owned by its neighbors, and eavesdrops the unused ones, to hear the newly joined nodes. A control packet includes the following information about the sending node: (a) both identifiers (ID and LID) (b) a bit-vector detailing which LIDs are occupied by the 1-hop neighbors, (c) status of each slot in the frame and (d) timestamp with local time.

In a network with N_F available frequencies, each node is assigned N_F transmission slots, and each transmission slot is assigned a different channel for data transmissions. On the other hand, each node is assigned a single receiving channel that it uses to receive data packets during its reception slots. Every node has one reception slot for each neighbor.

A TFMAC slot schedule is correct if for any given node n_i the following two conditions are satisfied:

- The transmission slot when n_i transmits on channel frequency f_j overlaps with reception slots of all its neighbors with assigned receiving frequency f_j.
- During the transmission slot when n_i transmits on channel frequency f_j, n_i is the only transmitter on frequency f_j in its two-hop neighborhood.

The first condition is necessary to provide a bidirectional data link between any two neighboring nodes, while the second one ensures collision-free communication.

B. Initialization Phase

Network initialization phase is divided into three time-separated stages: (a) neighborhood discovery, when nodes collect IDs of all nodes within their 2-hop neighborhoods, (b) LID and channel allocation stage, when each node is allocated a receiving channel and unique LID within the scope of its 2-hop neighborhood, and (c) time slot allocation stage, when each node is allocated one time slot for data transmission in the frame for every channel.

Essential requirement for all three stages is the ability to efficiently disseminate information, in a form of so called *initialization messages*, among nodes within 2-hop neighborhoods. During the first two stages, the protocol uses a simple, randomized (asynchronous) information dissemination scheme, and then, for the third phase, switches to deterministic (synchronous) TDMA-based transmission schedule. In randomized information dissemination scheme, each node transmits its initialization messages at randomly chosen times. When its transmission finishes at time t_i, the node schedules new transmission at time $t_{i+1}= t_i + T_d$, where T_d

is picked uniformly at random within the interval [0, T_{dmax}]. At time t_{i+1} the node sense medium and skip this transmission trail if medium is busy. In the deterministic information dissemination scheme, nodes are synchronized and transmit their initialization messages according to a collision-free transmission schedule. TDMA frame consists of N slots, and each node is allocated one transmission slot in the frame according to its LID. Two-hop uniqueness of assigned LIDs ensures conflict-free transmission.

During the node discovery stage, the initialization message contains ID of the transmitting node and IDs of all its currently discovered neighbors. Using the content of the received initialization message, the node updates its neighborhood list. In the second initialization stage, the initialization message contains not only IDs, but also LIDs and receiving channels of the transmitting node and all its neighbors with known LID and channel. Node chooses LID (so it's unique in its 2-hop neighborhood), when all its 2-hop neighbors with smaller ID have already chosen its LID. During the third stage, the squentiality is provided in the same manner as in the previous stage: the node chooses transmitting slots in the frame only if all of its 2-hop neighbors with smaller ID have already chosen their transmitting slots.

III. Performance Evaluation

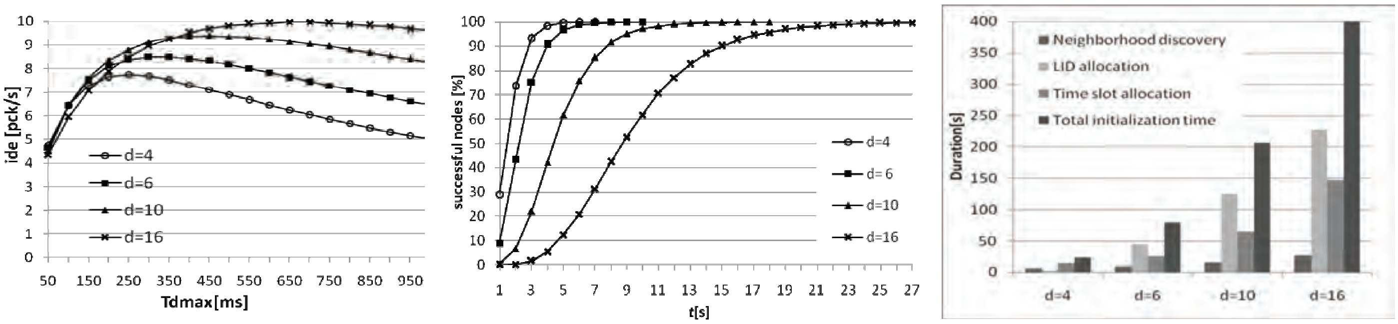
We implement enhanced TFMAC protocol in a custom WSN simulator build in C++, and conduct several experiments to evaluate its performances. All our evaluations are based on the simulation of the same WSN composed of 200 nodes randomly placed at fixed positions within an area of 100x100 m^2. The node density, which is defined as the average size of 1-hop neighborhood in the WSN, was varied indirectly, by varying radio transmission range. All nodes are equipped with single half-duplex transceivers with multi-channel capability. We used transceiver CC1100 as the hardware reference. Transfer rate of 20 Kbps is assumed, and packet length is fixed to the value of 64 bytes.

In the first set of experiments, we investigate the performances of the proposed three-stage self-organization mechanism in terms of time efficiency. The time needed to complete each individual stage of the initialization phase is primarily affected by the efficiency of the information dissemination scheme employed in the stage. This time is deterministic for the third stage, where TDMA-based information dissemination is used, and depends on the number of time slots per frame, only. Contrary, the duration of both neighborhood discovery and LID allocation stage is nondeterministic due to random nature of the asynchronous dissemination scheme used in these stages. This means that time periods reserved for these two stages, T_{ND} and T_{LID}, should be large enough to allow each node in the network to successfully finish its task with high probability. In order to estimate appropriate values of parameters T_{ND} and T_{LID} we rely on simulations. First we define the information dissemination efficiency (*ide*) as the average number of packets received per second at each node under asynchronous dissemination scenario. Intuitively, as *ide* is higher, the initialization process will be completed in less time and with less consumed power. The *ide* is influenced by the value of T_{dmax} which defines the rate at which nodes transmit their initialization messages. Choosing a small value of T_{dmax}, which means higher rate of initialization message transmission, would lead to a smaller *ide* due to frequent message collisions. With a large value of T_{dmax}, the probability of message collision will be small, but *ide* will be reduced again due to low rate of initialization message transmissions. The goal then is to choose T_{dmax} so as to maximize the *ide*. Fig 2(b) shows the *ide* as a function of T_{dmax} for four different node density values. As can be seen from Fig. 3(a), the optimal value of T_{dmax} depends on the node density (d), and ranges from 300 ms for $d = 4$ up to 750 ms for $d = 16$.

Fig. 3(b) provides insight on how the percentage of nodes that successfully finish neighborhood discovery stage changes as time progresses. Note that these simulations are carried out assuming the optimal value of T_{dmax} for each of four different node density values. According to Fig. 3(b), the adequate value of T_{ND} is highly dependent on node density. For example, when d=4, T_{ND} should be set to at list 5s to offer each node in the network to fully discover its neighborhood with high confidence. As node density increases, the time that should be reserved for neighborhood discovery stage increases also, reaching T_{ND}=25s for d=16.

Fig. 3(c) shows the estimated duration of each initialization stage as well as the total duration of the initialization phase for different node density values. As can be seen in Fig. 3(c), the LID allocation is the most time consuming initialization operation which takes more than 50% of total initialization time. This can be contributed to the relatively high complexity of the distributed LID allocation algorithm which requires several iterations of information exchange among 2-hop neighboring nodes. As a result of deterministic information dissemination, the time slot allocation stage lasts for about 30% less then LID allocation stage although basically the same algorithm is used in both stages. The neighborhood discovery is the least time consuming stage since it require one complete information exchange within each 2-hop neighborhood, only. As expected, the total initialization time is greatly dependent on node density and ranges from 25s, for low density network (d=4), up to 400s, for network with high node density (d=16).

The second set of experiments relates to the active phase of enhanced TFMAC protocol. Our focus in this study is on the benefits of multiple channels in terms of throughput and latency. As a performance metrics we adopt: the packet delay, and throughput. The packet delay is the average time, in seconds, from the arrival of a packet at the buffer of the source node to the arrival of the packet to the destination neighboring node. The throughput is defined as the ratio of the average number of packets that are successfully received at each node to the total simulation time. We run simulations by varying the following three traffic/network parameters: traffic load, number of available channels, and node density. Node traffic is statistically generated with packet inter-arrival time chosen from an exponential distribution with rate λ, i.e., with the average inter-arrival time 1/λ. The traffic load was varied by changing λ. Also, we assume the gossip traffic pattern, in which each node only communicates with its 1-hop neighbors.



(a) Information dissemination efficiency (b) Successfulness of the neighborhood discovery (c) Duration initialization stage

Fig. 3. Protocol performance in initializaton phase.

Fig. 4 and 5 show throughput and delay characteristics in network with node density *d*=6, when enhanced TFMAC protocol is configured with different number of channels. These simulation results confirm TFMAC's scalability in term of number of channels. When the number of channels increases from 1 to 8, the maximum throughput (i.e. the largest admissible traffic load yielding a finite average packet delay) increases from 1.68 pck/s to 5.46 pck/s. Also, with larger number of channels the protocol transfers packets between nodes with smaller delay. For example, when λ=0.8 pck/s, the average packet delay decreases from 0.53s to 0.32s when the number of channels increases from 1 to 8.

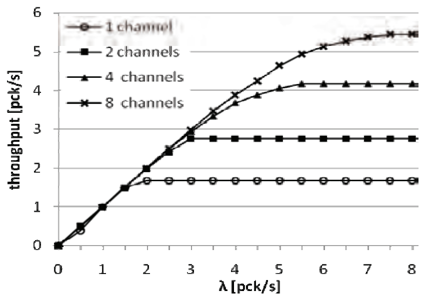


Fig. 4. Throughput over varying traffic load.

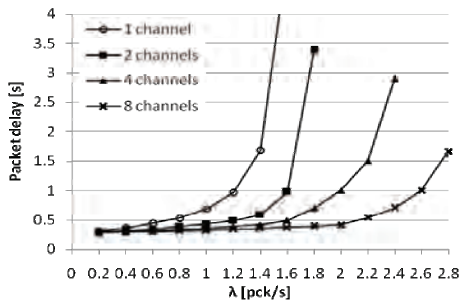


Fig. 5. Packet delay over varying traffic load.

IV. Conclusion

In this paper we have presented a schedule-based multi-channel MAC protocol designed for WSNs with high throughput requirements. The protocol adopts a channel/time slot allocation method of TFMAC, a multi-channel MAC protocol introduced in our previous work. In contrast to TFMAC, the protocol described in this paper is completely self-organizing in the sense that nodes are able to autonomously establish collision-free multi-channel schedules in a distributed manner. The self-organization is implemented as a three-state process that is carried out during initialization phase of the protocol. Simulation results show that (a) the initialization process completes in reasonable time even in high density networks, and (b) the multi-channel operation of the protocol greatly improves the throughput compared to a single-channel TDMA.

Acknowledgement

This work was supported by the Serbian Ministry of Science and Technological Development, project No. TR – 32009 - "Low-Power Reconfigurable Fault-Tolerant Platforms".

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] I. Demirkol, C. Ersoy, F. Alagoz, "Mac protocols for wireless sensor networks: a survey", IEEE Communications Magazine 44 (4) pp.115–121, 2006.

[3] "CC1100 Low-Power Sub- 1 GHz RF Transceiver", http://www.ti.com,

[4] M. Wang, L. Ci, P. Zhan, Y. Xu, "Multi-channel MAC Protocols in Wireless Ad Hoc and Sensor Networks," 2008 ISECS International Colloquium on Computing, Communication, Control, and Management, cccm, vol. 2, pp.562-566, 2008.

[5] G. Zhou, C. Huang, T. Yan, T. He, J. Stankovic and T.Abdelzaher, "MMSN: Multi-Frequency Media Access Control for Wireless Sensor Networks", In IEEE Infocom, April 2006.

[6] Y. Kim, H. Shin, and H. Cha, "Y-mac: An energy-efficient multi-channel mac protocol for dense wireless sensor networks," in Proceedings of IPSN '08, pp. 53–63, April 2008.

[7] M. Jovanovic and G. Djordjevic, "TFMAC: Multi-channel MAC Protocol for Wireless Sensor Networks", in Proc. of 8-th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services (TELSIKS) , pp. 23-26, Nis, Serbia, September 2007.