TFMAC: Multi-channel MAC Protocol for Wireless Sensor Networks

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

***Abstract* – In this paper we introduce TFMAC, a hybrid MAC protocol especially designed for wireless sensor networks in which each sensor node is equipped with a single half-duplex transceiver with multiple-frequency support. TFMAC incorporates multiple channels into a traditional TDMA scheme allowing different sensor nodes in a neighborhood to transmit on different channels simultaneously. Simulation results demonstrate that the proposed multiple-channel protocol provides higher maximum throughput and smaller average packet delay in respect to the basic single-channel TDMA scheme.**

***Keywords* – Wireless Sensor Networks, Medium Access Control, energy efficiency, Time Division Multiple Access.**

I. Introduction

A wireless sensor network (WSN) is a collection of large number autonomous sensor nodes that self-organize into a multi-hop wireless network. Nodes collaborate in order to perform sensing, computation, and data delivery in the execution of a common data acquisition task. WSNs are emerging technology with wide range of potential applications, e.g. environment monitoring, smart spaces, medical systems, military applications, general and civil engineering [1,2,3]. Improvements in hardware technology have resulted in low-cost sensor nodes, which are typically composed of a single chip embedded with memory, processor, and short-range transceiver.

Unlike other wireless networks, it is generally difficult or impractical to charge/replace exhausted batteries. That is why 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 will be 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 shared medium, i.e. 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 [4]. In general, the MAC protocol achieves energy efficiency by reducing the potential energy wastes, due to all, or some of the following reasons: idle listening, collision, overhearing, control-packet overhead, overemitting, and traffic fluctuations [5,6].

MAC protocols for WSNs can be divided into: contention-based and schedule-based protocols. In contention-based protocols, a common channel is shared by all nodes and it is allocated on demand [4,7,8]. A decentralized contention mechanism (like CSMA/CA, or RTS/CTS handshake) is employed to decide which node has the right to access the channel at any moment. These protocols are not complex and can be deployed easily. On the other hand, the overhead of control packets can be considerable in sensor networks where the data packets are not very large.

Schedule-based MAC protocols utilize distributed algorithms that schedule the channel access among nodes in a sensor network so that collisions are prevented before they occur. TDMA scheduling schemes are a major research area for contention-free WSNs [9,10]. Using a single frequency channel, TDMA divides the channel into time slots and allocates the transmission slots to network nodes. Slot scheduling avoids energy losses due to collisions, idle listening, and message overhearing. Control overhead is the only significant factor for energy inefficiency as they require setting up and maintaining schedules. However, in WSNs, because nodes do not exhibit high mobility, scheduling adjustment is needed infrequently and the corresponding overhead is negligible considering the long lifetime of WSNs.

Most of traditional MAC protocols for WSNs are designed for an environment where sensor nodes use simple, low-cost transceivers that can operate on a single frequency (channel), only. On the other hand, the current low-cost, low-energy transceivers, such as CC2420 radio [11], already supports multiple frequencies. Such transceiver can not transmit and receive at the same time, but it can switch its 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. One of few existing multi-channel MAC protocols, especially designed for WSNs, is MMSN [12]. MMSN protocol combines CSMA/CA with FDMA. In MMSN protocol, time is divided into multiple fixed-time beacon intervals. At the beginning of every interval packets are exchanged among nodes so that they can coordinate the assignment of appropriate channels for use in the subsequent time slots of that interval.

In this paper we present TFMAC, abbreviated from Time-Frequency MAC that exploits the existence of multiple channels and the ability of transceivers to switch between them quickly to increase network throughput. In order to

[1] Milica Jovanovic and Goran Djordjević are with the Faculty of Electronic Engineering, Aleksandra Medvedeva 14, 18000 Nis, Serbia, E-mail: {milicam, gdjordj}@elfak.ni.ac.yu.

provide conflict-free communication for data packets, the TFMAC divides time into a fixed number of time slots and allows each node to use different frequencies within different time slots to send data packets to its neighbors. Slot assignment is accomplished in a distributed way, through exchange of a limited number of controls messages during the contention slot at the beginning of each time frame.

II. TFMAC PROTOCOL DESCRIPTION

We consider multi-hop WSN, where no central entity exists to coordinate medium access and channel allocation. Each node is equipped with a single half-duplex transceiver. A transceiver can be tuned to different channels (frequencies), but can only use one channel at a time. The transmission range of a node determines the set of nodes it can communicate directly, which are also called its *neighbors*.

Like many other TDMA-based MAC protocols [10][9], TFMAC requires time synchronization. In TFMAC, time is partitioned in consecutive frames of fixed duration and each frame consists of a contention access period, followed by a contention-free period (Fig. 1). The contention access period, referred as *control slot*, is used to exchange control messages needed to maintain the protocol. All nodes must monitor the default frequency channel during this time period. The communication during the control slot is contention-based and uses the same scheme as the one used in 802.11 DCF [8]. The contention-free period, on the other hand, is used to transfer data between nodes. Each contention-free period is further divided into N_t equal sized *time slots* numbered from 0 to N_t-1. The duration of a time slot is long enough to accommodate transmission of one or more data packets. Transmissions in time slots are scheduled to avoid collisions.

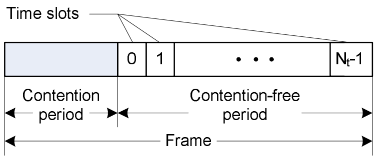


Fig.1. Frame structure

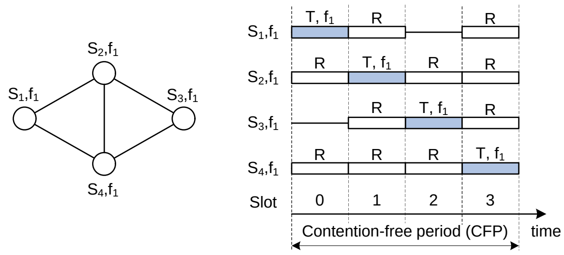
The timetable is the main local data structure that needs to be maintained at each node to keep current time slot usage information. For each time slot, the timetable contains an entry: (*i*, *t*, *f*), where i is the slot number, *t* is a slot type and *f* is the index of frequency used in that slot. Allowable slot types are: T, for transmission slot, R, for reception slots, and I, for idle slots.

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

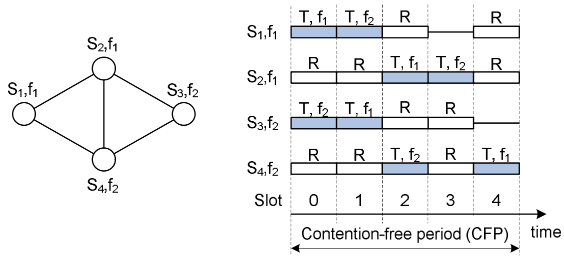
A TFMAC slot schedule is correct if for any given node n_i the following two conditions are satisfied:

- 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, ni 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.



(a) single-frequency schedule.



(b) two-frequency schedule.

Fig. 2. Examples of TFMAC's time slot schedules

Figures 2(a) and 2(b) shows single- and two-frequency TFMAC slot schedules for a simple WSN with four nodes, respectively. In the case of single-frequency (Fig. 2(a)), which corresponds to the classic TDMA scheme, four time slots are sufficient to establish conflict-free transmissions. As can be seen, each node is assigned a single receiving frequency, f_1, and a single transmission slot that it uses to send data packets to all its neighbors. Transmission slots of different nodes must not overlap, since all nodes in this simple WSN belong to the same two-hop neighborhood and use the same frequency. In two-frequency case (Fig. 2(b)), TFMAC assigns each node a single receiving frequency and two transmission slots, one for each of two available frequencies, f_1 and f_2. As can be seen, under some circumstances, the TFMAC allows concurrent transmissions to take place within the same two-hop neighborhood during the same time slot. For example, during slot 0, both nodes S_1 and S_3 are allowed to send data packets, since they use different frequencies, and communicate with disjoint sets of destination nodes. However, as the number of available frequency increases, the number of time slots per frame needed to accomplish correct slot schedule increases, too (5 for two-frequency vs. 4 for single-frequency example). Nevertheless, in a multi-frequency case, the total time that is allocated to nodes for data transmissions if typically larger than in a single-frequency case (2/5 CFP for two-frequency vs. 1/4 CFP for single-frequency scenario).

The TFMAC protocol consists of two aspects: frequency assignment and media access. The frequency assignment is used to assign available frequencies to nodes for data reception, while media access scheme establishes collision-free media assess schedules, in time and frequency domains, for individual nodes, in a distributed way.

TFMAC employs a simple frequency assignment scheme. Each node randomly chooses one of N_f frequencies as its receiving frequency and then broadcast its frequency decision to its neighbors, so that each node knows what frequency to use to transmit data packets to each of its neighbors.

After the nodes of a WSN are exchanged their receiving frequencies, the network enters a node activation period when timetables for individual nodes are set up such that transmissions from the nodes are conflict free in time and frequency domains. During this period, nodes are differentiated in terms of those that are already chosen their transmission slots (i.e. *active nodes*) and those that are not (i.e. *passive nodes*). Active node is allowed to participate in regular data traffic with other active nodes. On the other hand, passive nodes, although excluded from data communication, are obligated to monitor the control slot in order to exchange control/status messages needed to maintain the protocol.

When a node, say n_a, decides to activate itself, it starts the activation process with aim to select its transmission slots. To ensure conflict-free transmission slot selection, n_a must first acquire current schedules of all its neighbors (in a form of their timetables). The node n_a gather timetables of its active neighbors by first broadcasting a timetable-request packet in the control slot and then listening the media on default frequency for the duration of the current frame. Each active neighbor responds to a timetable-request packet by sending its timetable during its transmission time slot associated with the default frequency. Collecting timetables of passive neighbors is somewhat more involved since passive nodes can use the control slot for their transmissions, only. To get timetable of its passive neighbor n_i, activating node n_a sends an unicast timetable-request packet to n_i, while n_i promptly responds with timetable packet, all in the same control slot. Depending on the number of passive neighbors and the size of control slot, this timetable gathering process could last for several consecutive frames.

With timetables collected, activating node n_a is now able to carry out a conflict-free transmission slot selection. For each of N_f frequencies, n_a first finds the set of allegeable time slots, and then randomly selects one. Time slot t_i is allegeable for transmission at frequency f_j if the t_i is marked as idle slot in n_a's timetable as well as in timetables of all n_a's neighbors that use f_j as their receiving frequency. Such selection criteria guaranty that n_a will be able to send data packets, in a conflict-free manner, to all its neighbors that use f_j for data reception.

Node n_a finishes activation process by broadcasting, during control slot, a transmission-slot-announcing packet. This packet contains N_f pairs of (f, t), where f is the identifier of frequency that n_a uses in transmission slot t. Ali n_a's neighbors (active and passive) receive this packet and update their timetables accordingly. A neighboring node n_i with receiving frequency f_j is interested in the transmission slot that n_a uses for transmitting at frequency f_j. Thus, n_i identifies such slot in the received transmission-slot-announcing packet and mark its corresponding slot as reception slot.

In TFMAC, conflict-free transmission slot selection is not guarantied if two (or more) passive nodes in the same two-hop neighborhood start activation process simultaneously. Namely, it may happen that two concurrently activating nodes, with the overlapping sets of neighbors, chose the same time slot for transmission at the same frequency, which may result in collisions of their subsequent data transmissions at common neighbors. This can happen at network setup, when many nodes wake-up at same time, or when new nodes join the network. To reduce probability of collisions we introduce randomness in time between node synchronization with the network and starting its activation process. Moreover, a node postpones its activation as far as it is aware of an ongoing activation process. When taken together, these measures decrease likelihood of conflicting transmission slot selection at the great extent, but do not eliminate it completely. Fortunately, conflicting time slots can easily be detected by a neighboring node after it receives transmission-slot-announcing packet. A node which notices that a newly claimed transmission slot is already marked as reception slot in its timetable broadcasts the involved node ID number. That message will be overheard by the activating node, which will than back off and repeat activation process.

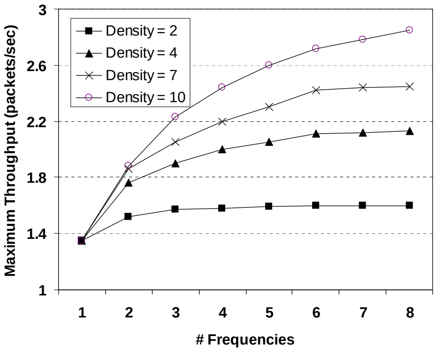
III. PERFORMANCE EVALUATION

We implement TFMAC protocol in a custom WSN simulator build in C++, and conduct several experiments to evaluate its performances. Our evaluation is based on the simulation of 200 nodes randomly placed in an area of 1000x1000 m^2. All nodes are equipped with single half-duplex transceivers with multiple-channel capability. Data rate of 20 Kbps is assumed, and data packet length is fixed to the value of 32 bytes. For TFMAC, CFP = 0.5 s and N_t=32, where CFP and N_t are the duration of contention-free period, and the number of time slots in a frame, respectively. Also, we assume the gossip traffic pattern, in which each node only communicates with its neighbors.

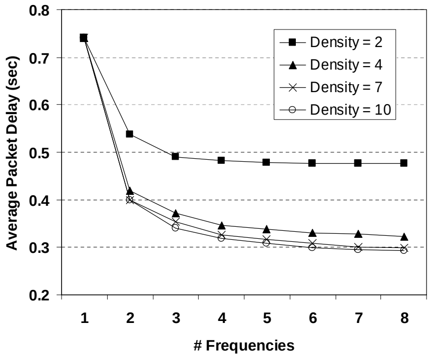
We run simulations by varying the following two traffic/network parameters: traffic load, and node density. Node traffic is statistically generated with packet inter-arrival time chosen from an exponential distribution with rate λ, i.e., with the average inter-arrival time $1/\lambda$. The traffic load was varied by changing λ. We define node density as the average number of nodes within radio transmission range, i.e. the average size of one-hop neighborhood. The node density is an important network parameter since it determines the number of reception slots that must be assigned to a node. The node density was varied indirectly, by varying radio transmission range. We tested node density values of 2, 4, 7 and 10.

Our focus in this study is on the benefits of multiple channels in TFMAC in terms of throughput and latency. As a performance metrics we adopt: the average packet delay, and the maximum throughput. Average 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 node. Maximum throughput is the largest admissible traffic load yielding a finite average packet delay.

Figures 3(a) and 3(b) show simulation results that relate to TFMAC's maximum throughput and average packet delay when different node densities and different number of available frequencies are utilized. In order to find a good estimate for the maximum throughput (Fig. 3(a)) we run a sequence of simulations by gradually increasing the traffic load, λ, until the observed average packet delay reaches the value of 1/ λ, i.e. to the point when the rate at which packets are generated equals the rate at which packets is delivered. That value of λ is then accepted as an estimate for the maximum MAC throughput. Packet delay characteristics in Fig. 3(b) is derived for the traffic load of λ=1.34 packets/sec, which corresponds to the maximum throughput of the single-frequency TFMAC.



(a) Maximum throughput



(b) Average packet delay

Fig. 3. Performance of TFMAC

In TFMAC with *k* available frequencies, each node is assigned *k* transmission slots that it can use to send *k* data packets to its *k* different neighbors during each frame. As *k* increases, the number of transmitted packets increases, too, until each node is able to send one packet per frame to each of its neighbors. Further increase in the number of frequencies has less effect on the TFMAC's performances since there are no more neighbors to send packets. This is the reason why the maximum throughput and average packet delay saturates by adding frequencies. For low node density (Density = 2), when the number of frequencies increase from 1 to 8, the maximum throughput increases from 1.34 to 1.6 packets/sec, while the average packet delay decreases from 0.74 to 0.48 sec. On the other hand, when the node density is higher (Density = 10), the maximum throughput that can be achieved with 5 frequencies is 2.89 packets/sec, while the average packet delay is 0.29.

IV. Conclusion

In this paper we have presented TFMAC, a media access control protocol for wireless sensor networks that allows exploration of modern low-cost and low-power radio transceivers' capability to dynamically switch their operating frequency. In essence, TFMAC is a multi-frequency extension of a basic TDMA scheme. In a TFMAC with *k* available channel frequencies, the nodes in the network are capable to choose their own receiving frequency as well as a set of *k* transmission slots (one for each frequency) for sending data packet to their neighbors. Simulation results show that the protocol improves both the maximum network throughput and average packet delay when compared to a single-frequency TDMA. In general, TFMAC is well suited for wireless sensor network applications that generate a variable, bursty local traffic exchange between neighbouring sensor nodes, such as event detecting or tracking applications.

References

[1] I.F. Akyldiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "Wireless Sensor Networks: A Survey", *Computer Networks*, Vol. 38, No. 4, pp. 393-422, March 2002.

[2] G.J. Pottie, W.J. Kaiser, "Wireless Integrated Network Sensors", *Communications of the ACM*, Vol. 43, No. 5, pp. 51-58, May 2000.

[3] Karl H. and Willig A., *Protocols and Architectures for Wireless Sensor Networks*, John Wiley and Sons, Ltd, Chichester, England, 2005.

[4] 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.

[5] Demirkol, C. Ersoy, f. Alagoz, "Mac Protocols for Wireless Sensor Networks: A Survey", *IEEE Communications Magazine*, vol. 44, iss. 4, pp. 115-121, 2006.

[6] K. Langendoen and G. Halkes, "Energy-Efficient Medium Access Control", *in Embedded Systems Handbook*, R. Zurawski, ed., CRC press, pp. 34.1-34.29, 2005.

[7] T. V. Dam and K. Langendoen, "An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks", in 1st ACM Conf. on Embedded Networked Sensor Systems (SenSys 2003), pp. 171-180, Los Angeles, CA, November 2003.

[8] IEEE 802.11 Working Group, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," 1997.

[9] V. Rajendran, K. Obraczka, and J. Garcia-Luna-Aceves, "Energy-efficient, collision-free medium access control for wireless sensor networks", in 1st ACM Conf. on Embedded Networked Sensor Systems (SenSys 2003), pp.181-192, Los Angeles, CA, November 2003.

[10] L. van Hoesel and P. Havinga, "A lightweight medium access protocol (LMAC) for wireless sensor networks", In 1st Int. Workshop on Networked Sensing Systems (INSS 2004), Tokyo, Japan, June 2004.

[11] CC2420 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver, available at http://www.chipcon.com

[12] 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.