Interference-Free TDMA Slot Allocation in Wireless Sensor Networks

Mário Nunes, António Grilo
INESC, Rua Alves Redol, Nº 9,
1000-029 Lisboa, Portugal
IST/UTL, Av. Rovisco Pais,
1096 Lisboa, Portugal
{mario.nunes, antonio.grilo}@inesc.pt

Mário Macedo
INESC, Rua Alves Redol, Nº 9,
1000-029 Lisboa, Portugal
FCT/UNL,
2829-516 Caparica, Portugal
mmm@fct.unl.pt

***Abstract*— This paper presents a distributed algorithm for efficient Time Division Multiple Access (TDMA) slot allocation in Wireless Sensor Networks, which is to be used in close association with a new TDMA-based MAC protocol called Latency-Energy Minimization Medium Access (LEMMA). Unlike most of the existent proposals, which try to guarantee interference-free allocation based on the *n*-hop criterion (which can only work with regular topologies), the presented algorithm bases its decisions on the received signal strength, allowing it to operate independently of the WSN topology. The performance of the proposed slot allocation algorithm was compared with centralized depth-first slot allocation using computer simulation. The results show that the proposed scheme is significantly faster, only paying the price of extra energy consumption during the initial network setup phase.**

Wireless Sensor Networks, MAC, TDMA, timeslot allocation

I. INTRODUCTION

Wireless Sensor Networks (WSN)s constitute a subset of ad-hoc networks – self-organized network that can be established with no need for a pre-existent communications infrastructure. WSN nodes (which can be either sensors monitoring a target area or actuators intervening with the environment), are interconnected by means of a wireless communications technology, eventually collaborating to forward the sensorial data hop-by-hop from the source node to the sink nodes (monitoring stations or Gateways), and to external networks (e.g., the Internet) and vice versa. Since WSNs usually require large numbers of WSN nodes in order to completely cover a target area, usually cheap and hence greatly limited in terms of processing, communications and energy. Although the overall WSN optimization must be addressed at each and every layer of the WSN protocol stack, the scope of the proposals presented in this paper are confined to the Medium Access Control (MAC) function – although the existence of some cross-layer collaboration with other layers (e.g., Routing) is assumed.

This paper presents a distributed scheme for timeslot allocation in TDMA-based MAC protocols for WSNs. The most innovative feature of the proposed scheme is that it bases all decisions for Spatial-TDMA slot allocation on the real interference experienced by the WSN nodes, instead of relying on topology-dependent heuristics such as the *n*-hop neighborhood criterion.

The proposed agorithm is part of the specification of the LEMMA MAC protocol, but can be easily adapted to other TDMA-based MAC protocols. Besides, while being suited for convergecast traffic patterns, LEMMA and its slot allocation scheme are capable to support multi-sink traffic patterns.

The rest of this paper is organized as follows. Section II presents the most relevant related work and the motivation for an interference-based TDMA slot allocation algorithm. Section III presents LEMMA and its distributed slot allocation scheme. Section IV analyses how LEMMA's slot allocation scheme is able to resolve conflicts in several interference scenarios. Section V presents the simulation results and finally section VI concludes the paper.

II. RELATED WORK

WSN MAC protocols can be broadly classified in two main families or their combination: Carrier Sense Multiple Access (CSMA), and Time Division Multiple Access (TDMA). Contention-based protocols like S-MAC [1], T-MAC [2] and D-MAC [3] present some specific sources of inefficiency such as idle listening, collisions, message overhearing and control packets overhead. TDMA protocols, on the other hand, are well suited to avoid these problems, but require tight synchronization, have reduced flexibility to handle uneven or changing topologies, as well as irregular multi-hop communication patterns, often requiring complex and sometimes message intensive slot assignment algorithms to guarantee collision and interference free slot schedules. Still, TDMA MAC protocols are more efficient in most WSN scenarios where nodes are static and the topology and traffic patterns are stable. Since the work presented in this paper is exclusively related with TDMA MAC protocols, they will constitute the focus of this section.

TDMA MAC protocols presents as much diversity. Several protocols have been designed for quick broadcast/convergecast,

---

The work described in this paper is based on results of IST FP6 project UbiSec&Sens. UbiSec&Sens receives research funding from the European Community's Sixth Framework Programme.

other for generic communication patterns. The greatest challenges are the spatial reuse of the time slots, interference avoidance, low latencies, and energy-efficiency.

SS-TDMA [4] is a TDMA protocol designed for broadcast/convergecast in grid WSNs. Its basic assumption is that the interference range is different from the communication range, and that their ratio (y) provides an estimation of the number of nodes in the interference range that can't have the same slot number. Configurations are established for the different kinds of grids, such as square and hexagonal, in terms of y, namely for the slot number repeat interval. The slot allocation process tries to achieve cascading slot assignments. Each node receives messages from the neighbors with their assigned slots. The receiving node knows the direction of an incoming message, and adds a value to the neighbor's slot number, in order to determine its own slot number. The added values depend on the direction of the message, the y value, and the kind of grid. Although being a distributed algorithm, it requires that each node is aware of its geometric position, limiting its applicability to grid topologies or systems where a localization service is available.

NAMA [5] is a more generic pattern TDMA protocol that tries to eliminate collisions dynamically: all nodes compute a common random function of the node identifier and of the time slot, and the node with the highest value is allowed to transmit in the slot. NAMA is based on a 2-hop neighborhood criterion (nodes at three hops of distance can reuse slots) for its decisions, and presents the additional drawback of being computational intensive. Its derivatives TRAMA [6] and FLAMA [7] also rely on the 2-hop neighborhood and do not implement cascading slot allocation.

L-MAC [8] uses a random slot assignment algorithm that also ensures that nodes at 2-hops distance do not use the same slot number, since nodes maintain a bit-map of the slot occupancy detected by the neighbors. Transmission in a slot has two parts: a control message, and a data message. The control message serves various purposes, namely for synchronization, routing, and to indicate the slot number of the transmitting node. All nodes receive the control message, but they return to sleep mode if the message is not intended for them. If a collision occurs in the transmission of the control message, the nodes are informed and those that tried to transmit in that slot must choose another slot. L-MAC was later extended to allow nodes to control more than one slot (AI-LMAC [9]).

Regarding TDMA slot allocation schemes, in [10] a set of algorithms is presented for TDMA slot assignment. There are two main algorithms – node based and level based – that are centralized, and another algorithm that is distributed. The algorithms base their slot assignment decisions on a conflict graph that is constructed from the original graph. The conflict graph includes the nodes that cannot transmit at the same time because they can ear each other transmissions or can interfere with the other transmissions. However, the paper does not specify any model to provide these relationships.

In [11] a centralized algorithm for TDMA slot assignment is presented. This is based on genetic algorithms and particle swarm optimization, but the algorithm is still based on an n-hop neighborhood criterion to avoid interferences in the timeslots. Although this paper presents promising results, the centralized nature of the algorithm limits its applicability.

Crankshaft [12] is a TDMA-like MAC protocol that uses a frame composed of both broadcast slots and unicast slots. In the former all nodes wake up and listen to incoming messages, and all nodes are allowed to transmit. The unicast slots are assigned to each node on a MAC address modulo n basis, where n is the number of unicast slots in the frame. Nodes can compete to transmit a message to a node in the adequate unicast slot, using a contention window where contention resolution is performed, and a message exchange window for actual message transmission. The receiver (and owner of the slot) polls the contention window for a preamble that precedes packet transmission. A mechanism is presented to allow more than one owner of a unicast slot to communicate. Although Crankshaft seems to be very energy-efficient, delay minimization is not a property of this protocol.

In [13] it is shown that more complete interference models are needed, namely based on the signal-to-interference ratio, other than the graph approach (e.g., n-hop neighborhood criterion), which is solely based on the actual communication links (i.e. on the signal-to-noise ratio).

It can be easily demonstrated that the n-hop neighborhood criterion is not suitable to some WSN topologies. Consider the WSN depicted on Fig. 1. Nodes A5 and B5 cannot communicate with each other, but can still interfere. Each node considers itself as located at a 10-hop distance from the other. A time-slot allocation that is based exclusively on the usual 2-hop neighborhood criterion might assign the same time-slot to A5 and B5, leading to poor link quality in both nodes.

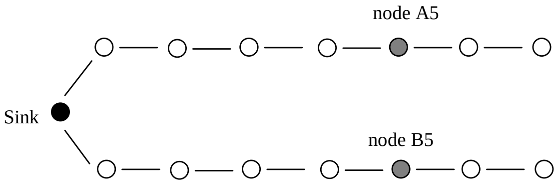

Figure 1. Example of a WSN topology where the n-hop neighborhood criterion does not work as desired. Nodes A5 and B5 are not within communication range, while they still interfere severely with each other.

Some other TDMA MAC protocols depart from the n-hop neighborhood criterion, but rely on different node transmission powers in order to avoid or identify and reduce possible interferences on the slots (see, [14][15][16][17][18]). But this design option can be considered a major drawback, because it limits the transmission range of the nodes.

III. LEMMA

Latency-Energy Minimization Medium Access (LEMMA) is a Spatial TDMA MAC protocol for WSNs, which was developed aiming to minimize latency in the most energy-efficient way, taking into account compatibility with the IEEE 802.15.4 and Zigbee specifications.

LEMMA is divided in two parts, each addressing a different phase of WSN operation: time-slot allocation during the initialization phase (transitory state) and interference-free data transmission (steady-state). LEMMA does not define how and when the system transits from the transitory state to the steady state, relegating these decisions to WSN management functions. Additionally, LEMMA does not address the establishment of routes, though it assumes that WSN routes are established a priori in a tree topology that could support either upstream convergecast or downlink multicast.

LEMMA's slot allocation mechanism – the main focus of this paper – is generic and may be integrated with other Spatial TDMA MAC protocols. The steady-state operation will only briefly described to allow a better understanding of the timeslot allocation procedures, since it constitutes the main subject of another paper.

A. *Steady-state Protocol*

LEMMA seeks to minimize latency in the most energy-efficient way, assuming a sporadic convergecast traffic pattern and corresponding logical tree topology, typical of alarm-driven WSNs – though it can support multi-sink operation with several logical trees running in parallel. These goals are achieved coupling a near-optimal cascading TDMA slot allocation (low latency) with a very low duty cycle (energy-efficiency). The TDMA frame is divided into a number of time-slots. In the case of convergecast, the slot allocation algorithm tries to assign the timeslots with the highest number to the nodes that are directly connected to the sink node, and a similar policy is followed at each level of the tree, resulting in a cascading assignment that minimizes the latency required to transmit one packet from source to sink. The slot allocation algorithm assures that the access to each time-slot is interference-free, assigning each slot to only one node within the interference vicinity, but it takes advantage of spatial slot reuse outside of that vicinity. This is exploited wherever possible to minimize end-to-end latency even more. In fact, for the convergecast tree depths expected in practical WSNs, a packet transmitted to the air by any source will usually arrive at the sink within one TDMA frame period.

An illustrative example is presented in Fig. 2, the numbers inside the time-slots represent the assignment for data transmission. In this example, a TDMA frame of duration equal to 1 s is divided in 200 time-slots of 5 ms. In this simple example each node is assigned one slot only, which it uses to transmit data to its parent in the convergecast tree, with the sink at node 0, but LEMMA supports multi-slot assignments as well. The example considers that nodes 5, 6 and 7 do not interfere with nodes 3 and 4. This allows the same slots to be assigned to 3 and 5 and to 4 and 6.

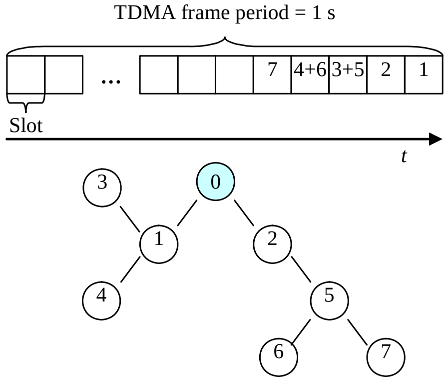

Figure 2. TDMA slot assignment for low-latency low-duty-cycle operation.

B. *Interference-free TDMA Slot Allocation*

The time-slot allocation algorithm has the goal to achieve the non-interfering cascading assignment described in the previous section, so that source-to-sink transmission latency is minimized. A possible mechanism to achieve an ideal slot assignment is to employ a centralized depth-first algorithm. However, this would lead to a long WSN setup delay. On the other hand, distributed interference-free slot allocation is not simple to achieve in a complex WSN topology operating in a real environment, subject to propagation and interference phenomena. This is the reason why many TDMA MAC protocols rely on rules of thumb such as the *n*-hop neighborhood criterion, or assume specific physical topologies to achieve near-optimal slot assignments (e.g., SS-TDMA).

The distributed time-slot allocation strategy proposed in this paper follows an interference-based approach, basing slot assignment and re-assignment decisions on the link quality directly experienced by the nodes, without assuming a direct relation between the interference and the number of hops. It is also independent of the geographical location of WSN nodes.

In the LEMMA TDMA frame, slot 0 is a special random access slot used for signaling, including slot allocation and synchronization. All the other slots can be dynamically assigned to the WSN nodes for data transmission. Each node keeps a slot occupation table, in which all slots are initially marked free. During the allocation phase, each data slot is assumed to be divided into a number of Allocation Check Windows (ACWs) of fixed size, within which nodes will compete for slot allocation.

The slot allocation starts at the sink and comprises the following steps performed at each level of the tree:

I) Operations in slot 0

1) The parent node multicasts a Request To Assign message in slot 0 (designated RTA0), indicating the slots that it shall attempt to allocate in order that each of its children can send data to it. The chosen slots will correspond to the highest free values in the TDMA frame, one slot for each child. Of course, these slots must be different from those previously assigned to other children. Parameters for this message include:

source address, destination addresses of the children (*i*), together with the respective proposed time slot (TS_i);

2) Each child node checks whether slot TS_i is already marked as occupied (due to previously detected allocations, transmissions or interference in this slot):

   a) If the slot *TSi* is not already marked as occupied, the child node contends to send a Clear To Assign (CTA) message back to the parent so that the parent can resume the slot reservation process and the neighbors are informed that the slot is now occupied. It also schedules itself to wakeup in slot TS_i.

   b) Otherwise it will not respond to the parent;

3) The parent node collects the CTA responses from its children[1]:

   a) If a CTA response was received from child node *i*, the parent node schedules itself to wakeup in slot TS_i and the algorithm proceeds to step 4);

   b) If it did not receive a CTA from child node *i*, it chooses the next free slot and the slot allocation to child *i* is renewed in slot 0 of the following TDMA frame;

II) Operations in slot TS_i

4) If a CTA response was received from child node *i*, the parent node sends another RTA message in slot TS_i of the same TDMA frame to child node *i*. This message is sent using the backoff mechanism of the CSMA/CA MAC protocol (e.g. using the standard backoff window) in unicast mode (since only child *i* is now supposed to be listening);

5) Assuming that the RTA0/CTA exchange was successful, child *i* awakes in slot TS_i and listens to the medium (carrier sense mechanism of the MAC protocol):

   a) If no activity is detected, then it does nothing;

   b) If it receives a RTA message from its parent and until that instant it has not detected any activity in the slot, it responds to the parent with a CTA message and waits for the beginning of the next ACW, returning to the beginning of step 5;

   c) If it receives a RTA message from its parent but has previously detected activity in the slot (messages or interference) in the current ACW, it responds with a Not clear To Assign (NTA) message;

6) The parent node waits for the reply from child i:

   a) If the parent receives a CTA message, it waits for the current ACW to expire and goes back to step 4, renewing the RTA/CTA handshake until there is no ACW left in slot TS_i. In this way, contenders for the slot are successfully eliminated in each ACW;

   b) If the parent does not receive any answer from the child until the end of the slot, it tries again in the following ACW, or, if this is the last ACW, it gives up (preparing itself to repeat the process in the following TDMA frame, randomly selecting a new TS_i slot from the next *m* free slots, where *m* is a small integer, set to 1 in the simulations presented in this paper).

   c) If the parent detects any other message or interference from other nodes during the backoff window, it considers that the slot cannot be allocated and gives up (preparing itself to repeat the process in the following TDMA frame, randomly selecting a new TS_i slot from the next *m* free slots).

7) If the parent receives a CTA as the answer to the RTA message issued in the last ACW of slot TS_i, it considers that slot to be successfully allocated;

8) When there is no data traffic, the RTA/CTA exchange is periodically repeated in the respective slot in order to mark it as allocated.

The time slot allocation process terminates when all nodes get a free slot. In case any interference is detected during steady-state operation, a similar re-allocation process is started, involving only the affected nodes and tree branches.

Fig. 3 shows a simplified message diagram of the described algorithm, in which the parent tries to allocate slot TS_i to node *i* (success) and slot TS_j to node *j* (failure).

It should also be noted that the slot allocation packets are short, allowing several ACW in one slot. Of course, the effectiveness of the conflict resolution increases with the number of ACWs that can fit in one slot.

The slot allocation algorithm is exactly the same in case of multi-slot operation. In this case, in order to optimize slot assignments, the allocation of an additional slot chain in LEMMA should only start after the previous slot chain allocation is finished in the vicinity of the node.

---

[1] This description assumes a positive confirmation model, in which slot allocation is attempted only for children whose CTA response was received by the parent. Depending on the number of children and the length of the slot, a negative confirmation model may be more advantageous, in which there are no CTA responses to the broadcast RTA0 frame, it being sufficient for the parent not to receive a Not clear To Assign (NTA) response from child i in order to proceed with the allocation attempt for the respective slot TS_i. The confirmation model can be indicated by the parent by means of a special flag in the RTA frame.

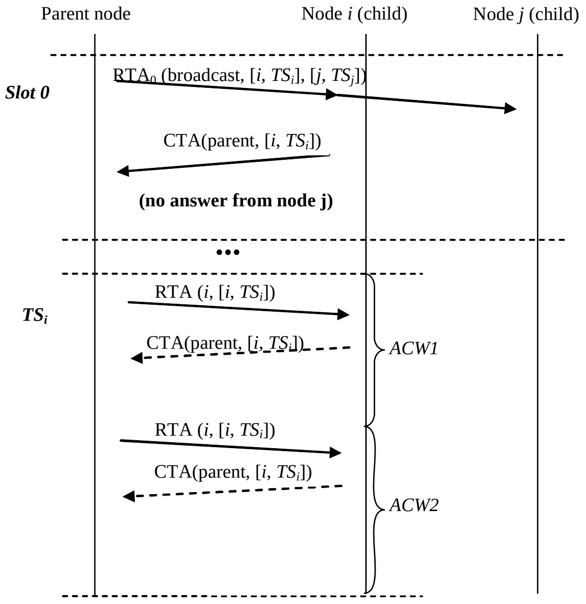

Figure 3. Message diagram of the slot allocation algorithm.

IV. INTERFERENCE SCENARIO ANALYSIS

This section illustrates the robustness of the proposed timeslot allocation protocol in different scenarios of interference between distant WSN nodes. These constitute only a basis upon which more complex scenarios can be drawn. Since the RTA/CTA/NTA exchanges are able to solve hidden-terminal problems, the selected scenarios focus exclusively on interference situations. This means that in all the pictures that describe the scenarios it is assumed that the parent nodes (P*x*) can only send/receive frames to the respective children (C*x*), while nodes belonging to different parent-child pairs can only interfere (the circles represent the interference ranges centered on the respective nodes, which is in all cases assumed to be greater than the communications range). In the scenario analysis, each parent-child pair tries to allocate a common timeslot by means of an RTA/CTA exchange in the selected timeslot. It is assumed that the RTA0/CTA exchange has already succeeded for the involved parent-child pairs.

A. Parents interfering with children of each other but not between themselves.

This scenario is depicted in Fig. 4. The following situations may occur:

1) Node P1 transmits the RTA first, and C1 responds with CTA. After P2 transmits the RTA, C2 will respond with NTA, prompting P2 to quit.

2) Node P2 transmits the RTA first, and C2 answers with CTA. After P1 transmits the RTA, C1 will respond with NTA, prompting P1 to quit.

3) If P1 and P2 transmit at the same time, there is a collision sensed both by C1 and C2, and thus none of them will respond, prompting P1 and P2 to retry in the following ACW.

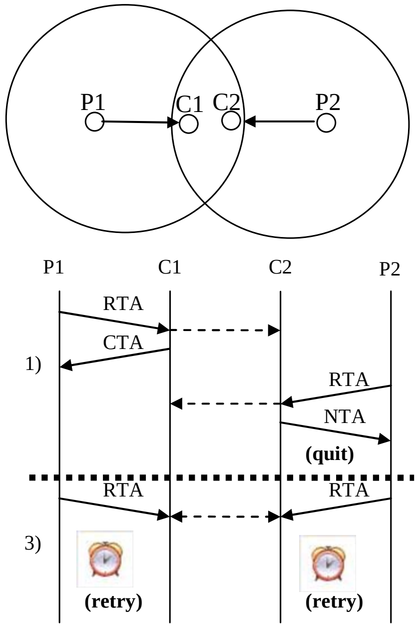

Figure 4. Interference scenario A.

B. Parents do not interfere, but one of the parents interferes with the other parent's child

This scenario is depicted in Fig. 5. The following situations may occur:

1) P1 transmits the RTA first, and C1 answers with CTA. When P2 transmits the RTA, C2 will respond with NTA, prompting P2 to quit.

2) P2 transmits the RTA first and C2 answers with the CTA, received correctly by P2. P1 senses the transmission from C2 and quits, not issuing his RTA.

3) If P1 and P2 transmit at the same time, there is a collision sensed by C2, while C1 receives the RTA correctly and responds with CTA. P1 keeps re-transmitting the RTA frame in the following ACW, trying to mark its ownership. Since P2 receives no response, it will also contend in the next ACW.

4) P2 transmits the RTA first and C2 answers with the CTA, received correctly by P2. If P1 transmits the RTA at the same time that C2 transmits the CTA, C1 will not be affected and will respond with his own CTA, allocating the same slot. Nevertheless, the RTA/CTA exchanges will be repeated in another ACW (or another superframe in case this persists in the last ACW of the current slot) and sooner or later case 1), case 2) or case 3) will occur, forcing one of the pair to select another slot.

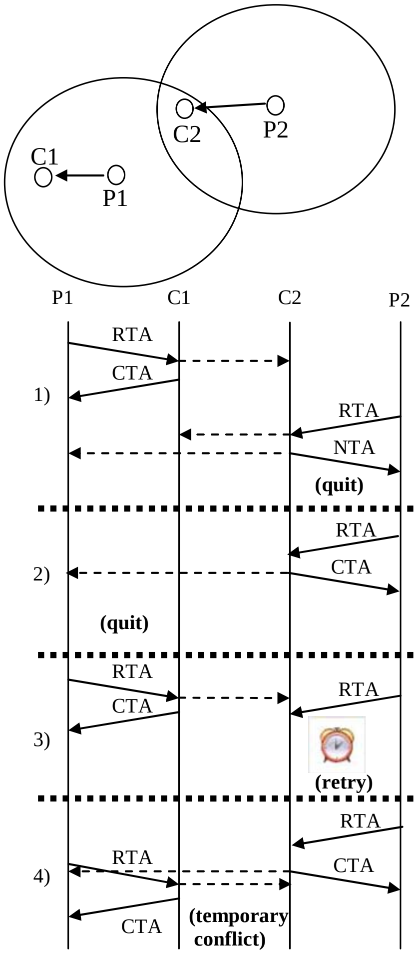

Figure 5. Interference scenario B.

C. Parents interfere between themselves, but not with the respective children

This scenario is depicted in Fig. 6. The following situations may occur:

1) P1 transmits the RTA first, and C1 answers with CTA. P2 senses P1's RTA transmission and quits.

2) P2 transmits the RTA first, and C2 answers with CTA. P1 senses P2's RTA transmission and quits.

3) P1 and P2 transmit the RTA frames at the same time. Both C1 and C2 will receive these frames correctly and respond with CTA, allocating the same timeslot. Nevertheless, the RTA/CTA exchanges will be repeated in another ACW (or another superframe in case this persists in the last ACW of the current slot) and sooner or later case 1), or case 2) will occur, forcing one of the pair to select another slot.

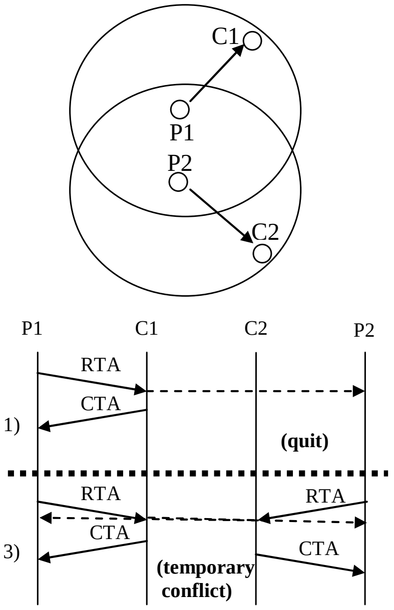

Figure 6. Interference scenario C.

D. Parents interfere between themselves and with the respective children

This scenario is depicted in Fig. 7.

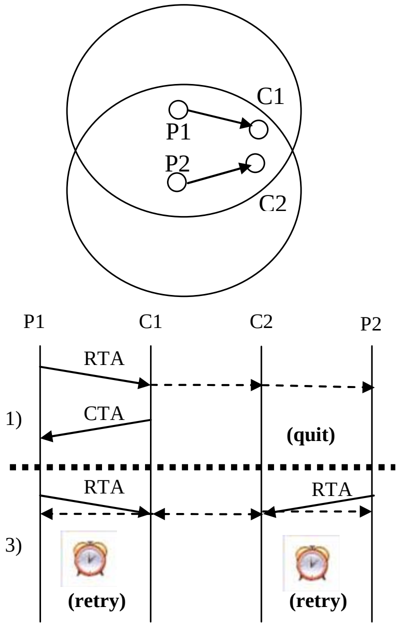

Figure 7. Interference scenario D.

The following situations may occur:

1) P1 transmits the RTA first, and C1 answers with CTA. P2 senses P1's RTA transmission and quits.

2) P2 transmits the RTA first, and C2 answers with CTA. P1 senses P2's RTA transmission and quits.

3) If P1 and P2 transmit at the same time, there is a collision sensed both by C1 and C2. Since P1 and P2 receive no response, they both retry in the following ACW.

E. Parents interfere between themselves and one of them interferes with the other's child node

This scenario is depicted in Fig. 8

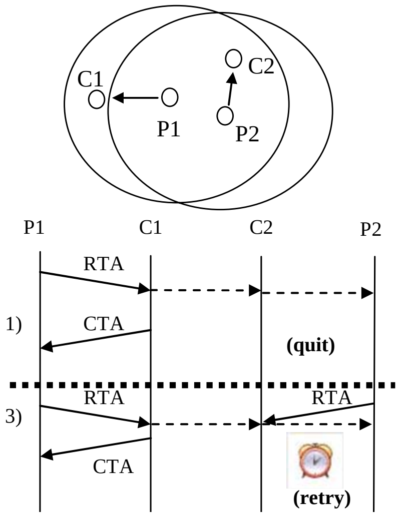

Figure 8. Interference scenario E.

The following situations may occur:

1) P1 transmits the RTA first, and C1responds with CTA. P2 senses P1's RTA transmission and quits.

2) P2 transmits the RTA first, and C2 responds with CTA. P1 senses P2's RTA transmission and quits.

3) If P1 and P2 transmit at the same time, there is a collision sensed by C2, while C1 receives the RTA correctly and responds with CTA. P1 keeps re-transmitting the RTA frame in the following ACW, trying to mark its ownership. Since P2 receives no response, it will also contend in the next ACW.

F. Child nodes interfere between themselves but not with the respective parent nodes.

This scenario is depicted in Fig. 9. The following situations may occur:

1) The exchange between P1 and C1 occurs first and C2 is able to sense C1's CTA response. In case C2 receives an RTA from P2 later on, it will respond with NTA.

2) The exchange between P2 and C2 occurs first and C1 is able to sense C2's CTA response. In case C1 receives an RTA from P1 later on, it will respond with NTA.

3) P1 and P2 transmit the RTA frames at the same time. Both C1 and C2 will receive these frames correctly and simultaneously respond with CTA, allocating the same timeslot. Nevertheless, the RTA/CTA exchanges will be repeated in another ACW (or another superframe in case this persists in the last ACW of the current slot) and sooner or later case 1), or case 2) will occur, forcing one of the pair to select another slot.

4) P1 sends an RTA and C1 responds with CTA, at the same time that P2 sends it RTA. C2 will experience a collision. P1 keeps re-transmitting the RTA frame in the following ACW, trying to mark its ownership. Since P2 receives no response, it will contend again in the next ACW.

5) P2 sends an RTA and C2 responds with CTA, at the same time that P1 sends its RTA. C1 will experience a collision. P2 keeps re-transmitting the RTA frame in the following ACW, trying to mark its ownership. Since P1 receives no response, it will contend again in the next ACW.

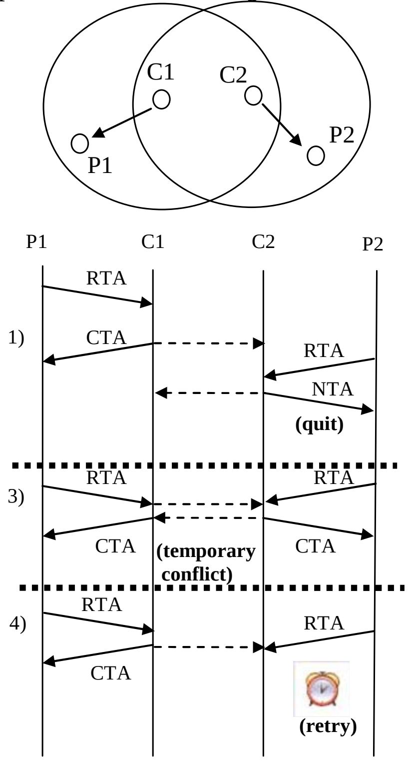

Figure 9. Interference scenario F.

From this analysis it can be concluded that the proposed algorithm is able to detect and gradually eliminate conflicts in Spatial TDMA slot allocation, even without any knowledge about the physical topology or any additional assumptions like the *n*-hop criterion.

V. RESULTS

The performance of LEMMA's distributed slot allocation (L) was evaluated comparatively with Depth-First (DF) allocation considering a square grid topology (in fact there is a slight jitter in the node deployment positions) where each node can communicate with its immediate vertical and horizontal neighbors and the interference range is approximately 2 hops. The TDMA frame period was 1 s (68 time slots) and each slot in LEMMA is configured to bear 3 ACWs. Results for the allocation of 1 slot are presented in Tab. 1, including the total duration of the slot allocation procedure, the size of the range occupied by the allocated timeslot chain, the number of slot allocation collisions that occurred (and were resolved) in the process and the total number of control messages that were transmitted during the slot allocation procedure. These results constitute the average over 10 runs.

TABLE I. COMPARISON BETWEEN LEMMA AND DEPTH-FIRST SLOT ALLOCATION SCHEMES.

Number of nodes	**Slot allocation duration (TDMA cycles)**		**Tree depth (hops)**		**Timeslot range**		**Allocation collisions**		**Control messages**	
	L	***DF***	***L***	***DF***	***L***	***DF***	***L***	***DF***	***L***	***DF***
4	2	4	2	2	3	3	2	0	8	6
9	4	9	4	4	7	6	6	0	22	16
16	6	16	6	6	11	9	13	0	43	30
25	8	25	8	8	15	12	21	0	69	48
36	10	36	10	10	19	15	32	0	102	70
49	12	49	12	12	23	18	44	0	140	96
64	14	64	14	14	27	21	59	0	185	126
81	16	81	16	16	31	24	75	0	235	160
100	18	100	18	18	35	27	94	0	292	198

Due to its distributed nature, LEMMA allocates a slot range that is not optimal compared with Depth-First, featuring a few gaps in the timeslot chain (which results in slightly higher source-to-sink delays). It also requires more messages to be exchanged due to retransmissions following interfering allocation attempts (note that this only applies during the WSN setup phase). However, it significantly reduces the time required for timeslot allocation, which in LEMMA is proportional to the depth of the tree (it should be noted that in Depth-First it is proportional to the number of nodes).

VI. CONCLUSIONS

This paper has presented a distributed algorithm for efficient Time Division Multiple Access (TDMA) slot allocation in Wireless Sensor Networks, which basis slot allocation decisions on the interference physically experienced by the WSN nodes, instead of relying on the standard *n*-hop neighborhood criterion. In this way, the proposed scheme is able to cope with irregular node deployments. The proposed scheme is part of the specification of the LEMMA MAC protocol, governing its operation during the WSN setup phase. However, it can be easily adapted to work with other convergecast tree TDMA MAC protocols.

Performance evaluation of LEMMA was accomplished using computer simulation. The distributed LEMMA slot allocation scheme proved to be significantly faster than Depth-First slot allocation, at the expense of only a slightly greater timeslot spreading and extra control message transmissions (and hence higher energy consumption) limited to the WSN setup phase.

Future work will focus on the improvement of LEMMA's slot allocation algorithm to handle topology changes and node mobility in a more efficient way. Another possible upgrade concerns the direction of timeslot allocation. Although the current downward distributed algorithm is significantly faster than centralized depth-first slot allocation, the authors envisage that upward time-slot allocation starting at the leaves of the convergecast tree can lead to better results.

REFERENCES

[1] W. Ye, J. Heidemann, D. Estrin, "An Energy-Efficient MAC protocol for Wireless Sensor Networks", in Proceedings of the IEEE Infocom, pp. 1567-1576, New York, June 2002.

[2] T. van Dam, K. Langendoem, "An adaptive energy-efficient MAC protocol for wireless sensor networks", in 1st ACM Conference on Embedded Networked Systems (SenSys 2003), pp. 171-180, Los Angeles, USA, November 2003.

[3] G. Lu, B. Krishnamachari, C. S. Raghavendra, "An Adaptive Energy-Efficient and Low-Latency MAC for Data Gathering in Wireless Sensor Networks", in Proceedings of the 18th International Parallel and Distributed Processing Symposium (IPDPS 2004), Santa Fe, NM, USA, April 2004.

[4] S.S. Kulkarni and M(U). Arumugam, "SS-TDMA: A Self-Stabilizing MAC for Sensor Networks", Sensor Network Operations, IEEE Press, 2005.

[5] L. Bao, J. J. Garcia-Luna-Aceves, "A New Approach to Channel Access Scheduling for Ad Hoc Networks," in the Proceedings of the Seventh Annual International Conference on Mobile Computing and Networking, ACM MobiCom 2001, Rome, Italy, July 2001.

[6] V. Rajendran, K. Obraczka, J. J. Garcia-Luna-Aceves, "Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks", in Proceedings of the ACM SenSys 2003. Los Angeles, USA, 2003.

[7] V Rajendran, J. J. Garcia-Luna-Aceves, K. Obraczka, "Energy-Efficient, Application-Aware Medium Access for Sensor Networks", in Proceedings of 2nd IEEE International Conference on Mobile Ad-Hoc and Sensor Systems, IEEE MASS 2005, Washington; USA, November 2005.

[8] L. van Hoesel, P. Havinga, "A Lightweight Medium Access Protocol (LMAC) for Wireless Sensor Networks", in Proceedings of the First International Workshop on Networked Sensing Systems (INSS04), Tokyo, Japan, June 2004.

[9] L. van Hosel, S. Chatterjea, P.J.M. Havinga, "A Low-Latency, Information-Centric Medium Access Protocol for Wireless Sensor Networks", the 1st International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP-2004), Melbourne, Australia, December. 2004.

[10] S. C. Ergen, P. Varaiya, "TDMA Scheduling Algorithms for Sensor Networks, Technical Report, Department of Electrical Engineering and Computer Science, University of California, Berkeley, July 2005.

[11] J. Mao, Z. Wu, X. Wu, "A TDMA scheduling scheme for many-to-one communications in wireless sensor networks", in Computer Communications, Vol. 30, Issue 4, pp. 863-872, Elsevier, February 2007.

[12] G. P. Halkes, K. G. Langendoen, "Crankshaft: An Energy-Efficient MAC-Protocol for Dense Wireless Sensor Networks", in Proceedings of the 4th European Conference on Wireless Sensor Networks (EWSN 2007), Delft, The Netherlands, January 2007.

[13]J. Grönkvist, A. Hansson, "Comparison between Graph-Based and Interference-Based STDMA Scheduling", the 2001 ACM Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc 2001), Long Beach, California, USA, Oct. 2001.

[14] W. R. Heinzelman, A. Chandrakasan, H. Balakrishnan, "Energy-Efficient Protocol for Wireless Microsensor Networks", in the Proceedings of 33rd Hawaii International Conference on System Sciences, Maui, Hawaii, USA, January 2000.

[15] W. B.Heinzelman, A. P. Chandrakasan, H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks, in the IEEE Transactions on Wireless Communications, Vol. 1, No. 4, pp 660-670, October 2000.

[16] Y.-C. Tseng, M.-Shiuan Pan, "Quick Convergecast in ZigBee/IEEE 802.15.4 Tree-Based Wireless Sensor Networks", The 4-th ACM International Workshop on Mobility Management and Wireless Access (MobiWAC 2006), Torremolinos, Spain, October 2006.

[17] S. C. Ergen, P Varaiya, "PEDAMACS: Power Efficient and Delay Aware Medium Access Protocol for Sensor Networks", IEEE Transactions on Mobile Computing, Vol. 5, No. 7, July 2006.

[18] G. Zhou, T. He, J. A. Stankovic, T. Abdlezaher, "RID: Radio Interference Detection in Wireless Sensor Networks", the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 20005), Miami, USA, March 2005.