

Minimizing Single TDMA Frame Sizes in Alarm-driven Wireless Sensor Networks Applications

Mário Macedo^{1,3}, Mário Nunes^{1,2}, António Grilo^{1,2}

¹ INESC-ID, Rua Alves Redol, N° 9,
1000-029 Lisboa, Portugal

² IST/UTL, Av. Rovisco Pais,
1096 Lisboa, Portugal

³ FCT/UNL,
2829-516 Caparica, Portugal

mmm@fct.unl.pt

{mario.nunes, antonio.grilo}@inesc.pt

Abstract Energy-efficiency and latency requirements in alarm-driven Wireless Sensor Networks often demand the use of TDMA protocols with special features such as cascading of timeslots, in a way that the sensor-to-sink delay bound can stay below a single frame. However, this single TDMA frame should be as small as possible. This paper presents a comparative study of timeslot allocation strategies that can be used to attain this goal. The Minimum Single Frame Size Problem is formulated, and the considered slot allocation algorithms are studied based on simulations. The results point to the conclusion that informed depth-first, coupled with a longest-path-first heuristic, can improve significantly the behavior of blind depth-first. Two centralized strategies are also simulated: a longest-paths-first, which allocates the branches by decreasing order of the length of the paths, and a largest-distances-first, which allocates the branches by decreasing distances to the sink that the paths can reach. It is also shown that a largest-distances-first strategy can achieve the smallest single frame sizes, and also the lowest variation of frame sizes. A distributed version of this algorithm (DIST-LDF) is presented, which obtains the same results of its centralized version.

1 Introduction

Wireless Sensor Networks (WSNs) are geographically distributed, self-organized and robustly networked micro-sensing systems that can be readily de-

ployed and operated in environments in which more conventional infrastructure-based systems and networks are impractical. WSN 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, and to external networks and vice versa.

In this paper, we are particularly interested to address critical alarm-driven WSN applications, such as surveillance of sensitive areas (e.g., intrusion detection and tracking). In this kind of WSN applications, traffic generation can be characterized as very sporadic, but the generation of an alarm report demands an immediate response to the event, which makes this kind of traffic very delay-sensitive. However, as WSNs devices have limited energy resources, low duty-cycles are also required. These two goals are usually contradictory, but among the MAC protocol families, TDMA protocols can provide low latency in the convergecast of data from the WSNs nodes to the sink, while still providing low duty-cycles. The quick convergecast of data from the nodes to the sink is usually achieved by constructing a routing tree routed on the sink node, and by ordering the timeslots in the path from a node to the sink, in such a way that the receiving slot(s) number(s) of a given node is lower than its transmitting slot number, while the slot distance between the two is kept as low as possible (a procedure that is called “cascading of timeslots”). On the other hand, low duty-cycle can be achieved by TDMA protocols, since each node only needs to be active during its receiving and transmission slots, while staying asleep for the rest of the TDMA frame. In order to maximize the overall goodput, there is also the need to reuse the same timeslots in different parts of the network, in a way that allows non-interfering transmissions to take place simultaneously.

While not being a pure TDMA protocol, the Data-gathering MAC (D-MAC) protocol, presented in [1], includes an adaptive duty-cycle, whose main purpose is to minimize the node-to-sink latency in convergecast networks. D-MAC uses staggered synchronization so that a data packet heard by a node, at one level of the tree, is transmitted to the next level in the following time period. The node is then allowed to sleep until the reception period for its level occurs. Nodes at the same level of the tree have to compete for timeslot access and may also interfere with nodes located in the same area. However, support of several sinks in D-MAC is troublesome

The use of TDMA for fast broadcast (a converse problem of convergecast) is a well-known subject (see, [2]). The authors show that the problem of determining optimal channel allocation for fast broadcasting is NP-hard. Two algorithms for tree construction, and slot assignment, are presented, namely a centralized version, and its distributed version. The distributed algorithm begins at the source node, and builds a spanning tree, such that each node has a slot number higher than its parent’s slot, but with the smallest possible value. Tree construction and slot assignment is performed depth-first, by means of passing a token to one node at a time, and by exchanging appropriate protocol messages with the neighbor nodes, in order to achieve conflict-free schedules. These protocols are also designed to

achieve spatial reuse of the slots, with relatively small TDMA frame sizes.

Another protocol that was designed to achieve TDMA conflict-free schedules is the DRAND distributed slot assignment protocol [3]. As the authors state, the problem of obtaining a minimum slot frame size is NP-hard. DRAND is not particularly suited for fast broadcast or fast convergecast, as slot assignment is random, but it is proved to achieve conflict-free schedules: DRAND assures that nodes in a 2-hop neighborhood do not use the same slot. DRAND also proved to have a message exchanging complexity of $O(\delta)$, where δ is the neighborhood of each node.

References [4] and [5] present two centralized algorithms, namely CTCCAA and CCA, which were specially designed to achieve low latencies in the convergecast process. CTCCAA proceeds with the tree construction and slot allocation processes in a breadth-first top-down manner, while CCA proceeds in a bottom-up manner from the leaves of the tree to the sink node. Both employ cascading slot assignment. The two algorithms differ in the way they establish the neighborhood of each node. These algorithms present the drawback of being centralized.

SS-TDMA [6] is a TDMA protocol designed for convergecast/broadcast applications. Its basic assumption is that the interference range is different from the communication range, and that the relation between them gives an estimation of the number of nodes within interference range that can't have the same slot number. In the slot assignment process, each node receives messages from the neighbors with their assigned slots. The receiving node knows the direction of an incoming message, and adds fixed values to the neighbor's slot number, in order to determine its own slot number. Although being a distributed algorithm, it needs a location service and topological knowledge about the networks, which limits its practical applicability. SS-TDMA also follows a cascading slot allocation, and a 2-hop slot allocation criterion.

More recently, the problems of building routing trees, and minimizing convergecast latency in ZigBee networks, were studied in reference [7]. The authors prove that the problem of obtaining a conflict-free slot assignment that minimizes the convergecast latency is also NP-hard. The distributed version algorithm is essentially a breadth-first tree construction and slot allocation protocol that is based on HELLO messages transmitted by the relay nodes. The main contributions of this protocol are the slot reassignment rules: the nodes that have more interfering neighbors, that stay closer to the sink, or that have a lower ID (identification number), have priority to choose a given slot that minimizes the latency. The interference avoidance procedure of this protocol is also based on a 2-hop neighborhood criterion.

Reference [8] presents a centralized algorithm and two distributed algorithms (namely, the DSA-AGGR algorithm – Distributed CCH for Data Aggregation). All the three algorithms are claimed to achieve a balance between depth-first and breadth-first in the tree slot allocation process. DSA-AGGR is designed to achieve low latency by means of cascading timeslots. DSA-AGGR begins to allocate slots from the sink to the leaves of the network tree, but each node is only eligible to al-

locate a slot if the following expression results in a value higher than 0.25 for the *color_score*:

$$color_score = \frac{2.ColoredOneHop + ColoredTwoHop}{2.NumberOneHops + NumberTwoHops} \quad (1)$$

Reference [9] presents a set of joint routing and slot assignment centralized algorithms that aim to achieve low latencies. However, the procedures have the drawbacks of being computational intensive, while the tree building process being centralized.

Finally, in reference [10] centralized algorithms are presented, which can be used to optimize the energy or the latency of the data collection process. These algorithms are hybrid, being based on genetic algorithms and particle swarm optimization. The centralized nature of these algorithms also limits their potential use.

2 Problem statement

Some of the works mentioned above are concerned with achieving short frame sizes. It is well known that for any greedy coloring algorithm, and also for any slot assignment algorithm, the worst-case of the number of the colors (or respectively the frame size) that are needed is equal to $\Delta+1$, where Δ is the maximum degree of a node, which is the maximum number of possible conflicting neighbors in the network.

As critical alarm-driven WSN applications should only report sporadic abnormal events, they do not periodically transfer bulk data to the sink. Therefore, achieving high throughputs for these networks is not a specific design requirement. Consequently, the problem addressed in this paper is not to obtain the smallest possible frame sizes, which depend on the maximum degrees of the networks. On the other hand, very small frame sizes also lead to high duty-cycles, and also imply that the transmissions from nodes that are placed away from the sink will potentially have to span several frames. This can originate different delay bounds for each node in the network tree. Finally, we are interested to have similar delay bounds for the alarms transmitted by all the network nodes. This requirement can be accomplished by transmitting the data always in a single TDMA frame, whatever the location of the node. However, since different scheduling algorithms may lead to different single TDMA frame sizes, we are also interested in algorithms that lead to the smallest possible single TDMA frame size in such a way that it is able to accommodate all the network nodes on it. We call this problem the Minimum Single Frame Size (MSFS) problem. To the best of our knowledge, we haven't seen this problem formulated before.

Intuitively, those algorithms will also lead to the lowest maximum slot distance

of the network (i.e., the lowest sensor-to-sink delay bound).

Some of the algorithms and protocols mentioned above assume that the building of the routing tree is done simultaneously with the slot allocation procedure, while in others the two processes are done separately. In this paper, it is assumed that the slot assignment procedure, of the network setup phase, is done after the routing tree construction, for example on top of an efficient contention MAC protocol like B-MAC [11].

3 Simulation Model and Scenario

Simulations were carried out considering a 100-node square grid physical topology, where the sink node was placed at the upper-right corner. A logical tree topology was assumed, with the sink being the root node, and each node communicating with a random neighbor, selected among those that were closer to the node, and that offered progress to the sink. In this way, each node was allowed to choose as parent either the node that is closest to it in the West direction, or in the North direction.

In accordance with most references in this subject, a simple free space propagation model was used. The path loss exponent was set to 2, and radio propagation irregularity was not considered [12] for the sake of simplicity.

The dimension of the grid square edges was set to the approximate value of communication distance. The interference graph was built based on a 2-hop neighborhood criterion, as it is customary in the related publications. As each slot was considered bi-directional, the links that were considered as interfering with parent-child communication were all the links established by the 2-hop neighbors of the parent node and of the child node, using the same timeslot.

Slot allocation begins at the sink (which allocates slots to its children) and proceeds independently down the network tree until the leaf nodes are reached. The simulations have considered different slot allocation strategies.

The dimension of each slot was configured to offer three transmission opportunities, in order to attain a sensor-to-sink delay bound of a single TDMA frame with high probability.

4 Slot Allocation Algorithms and Simulation Results

Centralized slot allocation algorithms are potentially more optimal and more predictable in terms of convergence, but present one drawback: they require that the nodes communicate their local topology to the sink (e.g., their neighborhoods, parents, etc.), which is a slow and communication intensive procedure. Additionally, they usually rely on theoretical propagation models to avoid assignment con-

flicts, being less adaptive to the space-time variability of realistic RF propagation environments. The distributed slot allocation algorithms can be particularly interesting because they do not require the sink node to know the network topology. Therefore, they are more scalable, flexible and adaptive, even if their convergence is less predictable and slower.

In this paper, for the matter of comparison, we did not consider the strategies referred in literature that are centralized, or that perform the building of tree simultaneously with the allocation process. All implemented strategies assume cascading of timeslots, and greedy slot allocation by each node (i.e., always choosing the highest possible slot number that is less than its parent's slot number).

The following slot allocation strategies were firstly considered: the classic depth-first (DF) and breadth-first (BF) [13], RANDOM, DSA-AGGR, and SS-TDMA.

The RANDOM strategy consists in selecting randomly and allocating any node whose parent node has already allocated also a slot. Therefore, the RANDOM strategy can descend the tree in several ways that fall between DF, and BF.

For the SS-TDMA slot allocation protocol, its y parameter was set to 2, meaning that the interference range was twice the communication range. In this way, if a node received a message from its Northern closest neighbor (its parent), it allocated a slot number equal to its neighbor's minus one; if it received a message from its Western closest neighbor, it allocated a slot number equal to its neighbor's minus $(y+1)$, or, in the case, minus 3. The simulations showed that SS-TDMA was able to allocate all slot numbers of a given frame without any unused slots, achieving also the spatial reuse of the slots, and implementing a 2-hop neighborhood interference avoidance criterion.

As the square grid had 100 nodes, and the nodes located at the top edge of the square have always its closest Western neighbor as parent, while the nodes located at the left edge of the square have always its closest Northern neighbor as parent, only 81 nodes can choose one node as parent, among its closest Western neighbor and its closest Northern neighbor. This means that 2^{81} different topologies can be generated, or 2.42×10^{24} different topologies. For each of these different topologies, there are also a huge number of different slot schedules that can be done by each slot allocation algorithm. For instance, DF can descend the tree visiting firstly different branches. These observations suggest that the number of different slot schedules that are possible is a very huge number, it being impossible to look at each of them to find the optimal TDMA frame size. Therefore, for each different slot allocation algorithm, we ran 10,000,000 simulations, each having as input one different random logical tree, and resulting in one different random slot schedule. Although this number is small in comparison with the number of all possible combinations of topologies and slot schedules, it was thought sufficient to assess how the different slot allocation algorithms behave with respect to the TDMA frame size that they can attain. The histogram of frequencies for the respective TDMA frame sizes was then built. Since the RANDOM strategy seems to generate a broader range of slot schedules, the set of simulations was raised to

100,000,000.

Fig. 1 shows the histograms for that set of slot allocation algorithms. The values corresponding to the RANDOM algorithm are the original divided by 10.

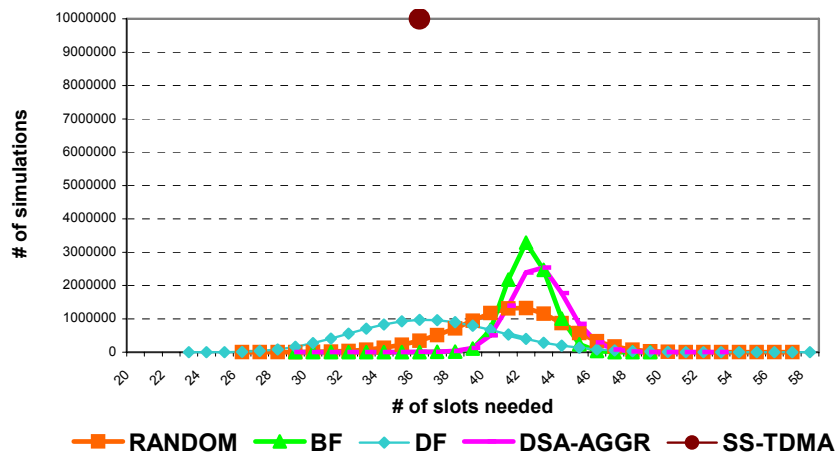


Fig. 1. Histograms of the number of simulations for each TDMA frame size, for the first set of slot allocation algorithms.

These results show that BF has a lower variance than DF, but DF achieves a lower average value, respectively 36.7 slots for DF, and 42.1 for BF (see, Table 1). DSA-AGGR does not behave as being a compromise between BF, and DF, with respect to finding low single TDMA frame sizes. However, the RANDOM algorithm behaves as expected: its histogram falls between those of BF, and DF. It is interesting to note that SS-TDMA achieves always the same number of slots, i.e., 36, whatever the logical topology. This result seems a little bit bizarre but the reason for it is simple to explain: the distances, in terms of slots, between the slot of a given node and the slots of its neighbors are constant, and they depend only on the directions of the neighbors. As the nodes of all simulated grids have always the same coordinates, the frames have always the same size. However, the number of slots that it is achieved with SS-TDMA is not particularly promising, as it is close to the average value that is obtained by DF, and substantially higher than the minimum values of DF.

DF can achieve a lower number of slots because it can descend first on larger branches. The cascading of timeslots in those larger branches results in a set of consecutive slots that can be reused in other adjacent upper and smaller branches of tree network. Inversely, if smaller branches are allocated first, those allocated slots cannot be used in larger adjacent branches, resulting in larger sensor-to-sink slot distances, and therefore larger TDMA frames. Note, however, that the slots that are used on an allocation of the longest branch first are not generally sufficient to color all the other network nodes, and that there is a need of some extra

slots: for instance, DF achieved a minimum of 23 slots in a given simulation, while the longest path had always a length of just 18 hops. On the other hand, BF achieves higher single TDMA frame sizes, because it allocates the nodes of all the branches at some level in the same round, meaning that when it tries to allocate a slot to a given node in a given branch, it finds more frequently slots that are already used. Therefore, in the allocation of one branch, namely the longest, the distances between the slots of a parent's node and the that of its child's node are generally longer, resulting in increased single TDMA frame sizes. DSA-AGGR seems to achieve the worst single TDMA frame sizes, as it tends to create hotspots of allocated nodes. Therefore, in the allocation of a given branch, namely the longest, the nodes tend to find more used slots, which are even occupied by nodes placed deeper in the tree, resulting in higher frame sizes.

These observations suggest that a longest-path-first strategy will lead to smaller TDMA frame sizes. In order to confirm this hypothesis, the next simulation considered a slot allocation algorithm that descends the tree in a longest-path-first manner when it has to make a decision of which path it chooses first, while allocating the other branches by backtracking in the same order of the depth-first strategy. This algorithm was designated depth-first-with-longest-path-first (DF-LPF). This strategy was proved to be advantageous over the simple DF, as is shown in Fig. 2. Average values decreased from 36.7 slots for DF, down to 27.9 slots for DF-LPF. These values have also resulted from 10,000,000 simulation runs.

Two centralized strategies were also investigated with the objective of achieving even smaller TDMA frame sizes.

The first centralized strategy (CENT-LPF, centralized longest-paths-first) allocates the branches in the descending order of their lengths, breaking the ties with priority for the paths that are situated deeper on the tree, and randomly if this rule is not enough to decide. The rationale for the first breaking ties rule is the same of the DF-LPF algorithm.

The other strategy (CENT-LDF, centralized largest-distances-first) allocates the branches in the descending order of the distances to the sink that the branches can reach, independently of their sizes. Breaking ties rules are the same of CENT-LPF.

Fig. 2 shows that there is a systematic improvement on the number of slots that are needed, when we successively consider DF-LPF, CENT-LPF, and CENT-LDF (which results are the same of its distributed version, DIST-LDF, that we describe later). Average values of the number of slots were respectively 27.9, 25.6, and 24.9, for these three slot allocation algorithms, as it can be seen in Table 1. This last allocation algorithm also produces the smallest range of values, among all the considered slot allocation algorithms. CENT-LDF presents slot numbers that range from 21 to 33 slots, while blind depth-first (DF) presented a much broader range, from 23 to 58 slots.

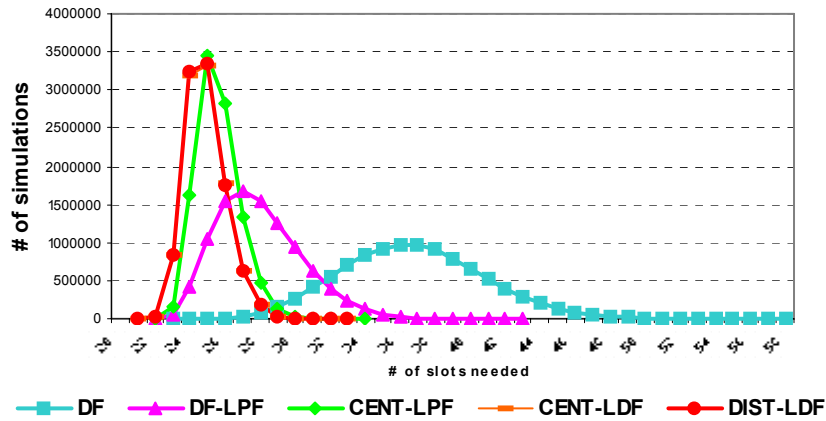


Fig. 2. Histograms of the number of simulations for each TDMA frame size, for the second set of slot allocation algorithms, showing improvements over simple depth-first (DF).

	Average (slots)	Minimum (slots)	Maximum (slots)
RANDOM	41	26	57
BF	42.1	29	49
DF	36.7	23	58
DSA-AGGR	42.7	29	53
SS-TDMA	36	36	36
DF-LPF	27.9	22	43
CENT-LPF	25.6	22	34
CENT-LDF	24.9	21	33
DIST-LDF	24.9	21	33

Table 1. Number of slots obtained by the allocation algorithms.

The explanation for these results is similar to that of the DF-LPF strategy: when we allocate firstly the branches that lead to larger distances from the sinks, we are cascading the slots in a such a way that the hop distances for the farthest nodes are minimized, while the nodes that are closer to the sink, and placed in adjacent branches, can reutilize the slots that are used in those larger branches, without demanding for many extra slot numbers.

With these results in mind, a distributed largest-distances-first strategy (DIST-LDF) was developed, which was proved to provide the same results as the centralized CENT-LDF. Its pseudo-code is listed in Fig.3.

At any node:

If there is more than one son:

For the son with the longest path to a leaf:

Allocate a slot for it, within the time interval Δt ;

For each other son i not in the longest path:

Allocate a slot, within the time interval =

$c * (\text{size-longest-path} - \text{size-path}(i)) * \Delta t$;

If there is only one son:

Allocate a slot for it, within the time interval Δt ;

Fig. 3. Basic pseudo-code of the distributed largest-distances-first (DIST-LDF) algorithm.

DIST-LDF descends the tree, allocating the slots, and when it has several different branches to allocate, it descends the branch that has a longest path firstly and immediately, while the others branches wait an amount of time that is proportional to the difference of the lengths of their paths and the length of the longest path. In this way, longest paths are allocated in advance, reserving slots for them before the smaller branches. Intuitively, in order for the allocations of longer distance branches not to be disturbed by the allocations of shorter branches, the former have to be scheduled sufficiently in advance.

Referring to Fig.3, Δt represents the time needed to allocate a slot in a real implementation distributed slot allocation protocol. Constant c has shown to generate the same results of the centralized algorithm, when it takes values greater or equal than 5. For lower values, DIST-LDF performance degraded progressively into higher values for the TDMA frame size. When constant c is lower than 5, some shorter distance branches do not wait for a sufficiently large delay, and begin to allocate slots for themselves, while longer distance branches are also allocating their slots, and these concurrent actions are not separated enough in space. Therefore, the shorter distance branches can allocate slots that can't be used by longer distance branches, resulting in larger TDMA frames. Or, in other words, longer distance branches are not scheduled sufficiently in advance. On the other hand, if we increase the value for c much beyond that of 5, we do not improve further the behavior of the DIST-LDF algorithm, because the allocation of longer distance branches is already being performed at distances in a way that they cannot be affected anymore by the allocations of shorter distance branches. Dimensioning of constant c for some specific network is, however, a matter for future research.

Although DIST-LDF presents the best results among the considered slot allocation strategies, it requires each node to know the length of the branches that are rooted at their children, which represents an important disadvantage. This information has to be propagated in the network tree, from the leaves to the sink, after the tree construction process, which is a procedure that results in a certain extra amount of overhead. Note, however, that the complexity of DIST-LDF, in terms of the number of visited nodes, seems to be intuitively smaller than that of DF, as DIST-LDF does not need to backtrack in the tree structure, when it full allocates a

sub tree. DIST-LDF is an always-forwarding allocating distributed discipline. Therefore, it is also expected that DIST-LDF can achieve smaller execution times than DF. However, these two last observations need to be assessed by future research.

5 Conclusions and Future Work

In most alarm-driven WSN applications (e.g., intrusion detection in restricted areas), traffic can be characterized as very sporadic, but the generation of an alarm report demands an immediate response to the event, which makes this kind of traffic very delay-sensitive. Low latencies and low duty-cycles can be simultaneously accomplished by using TDMA protocols. Cascading of the slots from a given node to the sink can result in low latencies, while TDMA protocols can achieve low duty-cycles, as each node need only to be awake in its slots. Since this kind of applications do not need high throughputs, and it is also desirable to have the same delay bound for all the nodes in the network, we defined a new problem, which is that of allocating slots for all the nodes of the networks, such data can be always transmitted in a single TDMA frame, whatever is the place of the node in the network. However, such unique TDMA must also have the smallest possible size (problem that we designated as the Minimum Single Frame Size – MSFS – problem).

In this paper, several TDMA slot allocation strategies were comparatively evaluated with respect to the goal of minimizing the single TDMA frame size. The simulation results have shown that a breadth-first slot allocation strategy behaves poorly than depth-first, and that an informed depth-first strategy, that visits the longest-path first, improves significantly the results when compared with blind depth-first. It was also shown that a largest-distances-first slot allocation algorithm will produce the smallest single TDMA frame sizes, and the smallest range of values, among all the scheduling algorithms that were considered. A distributed version (DIST-LDF) of this algorithm was implemented, which was able to obtain the same results as its centralized counterpart.

For future work, the authors intend to investigate if these disciplines also minimize the latency of the convergecast for the more general case of transmission done in several frames, and to develop new slot allocation algorithms that can even exceed the DIST-LDF algorithm performance.

Acknowledgments 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. Apart from this, the European Commission has no responsibility for the content of this paper.

The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

References

1. 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.
2. I. Chlamtac, S. Kutten; "Tree-based Broadcasting in Multihop Radio Networks", IEEE Transactions on Computers, Volume C-36, No. 10, Oct. 1987.
3. I. Rhee, A. Warrier, J. Min, L. Xu, "DRAND: Distributed Randomized TDMA Scheduling for Wireless Ad-hoc Networks", the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'2006), Florence, Italy, May 2006.
4. V. Annamalai, S.K.S. Gupta, L. Schwiebert, "On Tree-Based Convergecasting in Wireless Sensor Networks", in the Proceedings of the IEEE Wireless Communications and Networking (WCNC 2003), New Orleans, LA, USA, March 2003.
5. S. Upadhyayula, V. Annamalai, S.K.S. Gupta, "A low-latency and energy-efficient algorithm for convergecast in wireless sensor networks", in Proceedings of IEEE Global Telecommunications Conference (GLOBECOM '03), San Francisco, CA, USA, January 2003.
6. S.S. Kulkarni and M.(U.) Arumugam, "SS-TDMA: A Self-Stabilizing MAC for Sensor Networks", Sensor Network Operations, IEEE Press, 2005.
7. M.S. Pan, Y.-C. Tseng, "Quick convergecast in ZigBee beacon-enabled tree-based wireless sensor networks", Computer Communications, Vol. 31, Issue 5, pp. 999-1011, Elsevier, 25 March 2008.
8. K. L. Bryan, T. Ren, L. DiPippo, T. Henry, V. Fay-Wolfe, "Towards Optimal TDMA Frame Size in Wireless Sensor Networks", University of Rhode Island, Technical Report, TR-xxx, March 2007.
9. G. Lu, B. Krishnamachari, "Minimum latency joint scheduling and routing in wireless sensor networks", Ad Hoc Networks, Vol. 5, Issue 6, pp. 832-843, Elsevier, August 2007.
10. J. Mao, Z. Wu, X. Wu, "A TDMA scheduling scheme for many-to-one communications in wireless sensor networks", Computer Communications, Vol. 30, Issue 4, pp. 863-872, Elsevier, February 2007.
11. J. Polastre, J. Hill, D. Culler, "Versatile Low Power Media Access for Wireless Sensor Networks", in Proceedings of the 2nd ACM SenSys Conference, pp. 95-107, Baltimore, MD, USA, November. 2004.
12. T. Rappaport, "Wireless Communications: Principles and Practice", 2nd Edition, Prentice Hall, 2002.
13. T. H. Cormen, C. E. Leiserson, R. L. Rivest, "Introduction to Algorithms", The MIT Press, 2000.