Running head: SLOT ALLOCATION ALGORITHMS

SLOT ALLOCATION ALGORITHMS FOR MINIMIZING DELAY IN ALARM-DRIVEN WSNS APPLICATIONS

Mário Macedo, António Grilo, Mário Nunes

INESC-ID

Instituto de Engenharia de Sistemas e Computadores – Investigação e Desenvolvimento

Lisboa, Portugal

mmm@fct.unl.pt

{antonio.grilo, mario.nunes}@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 the duration of a single frame. However, this single TDMA frame should be as small as possible. The results presented in this paper, point to the conclusion that a largest-distances-first strategy can achieve the smallest single frame sizes, and also the lowest frame size variations. A quite simple distributed version of this algorithm is presented, which obtains the same results of its centralized version. Simulations also show that this discipline presents the best results in terms of sensor-to-sink slot distance, even if they require a few more slots than breadth-first in multi-frame scenarios.

KEYWORDS

WSNs, TDMA, Alarm-driven Applications, Slot Allocation Algorithms, Slot Distance, Cascading Minimum Single Frame Size Problem

SLOT ALLOCATION ALGORITHMS FOR MINIMIZING DELAY IN ALARM-DRIVEN WSNS

INTRODUCTION

Wireless sensor Networks (WSNs) are geographically distributed, self-organized and robustly networked micro-sensing systems that can be readily deployed and operated in environments in which more conventional infrastructure-based systems and networks are impractical, or cost-ineffective. WSN nodes are interconnected by means of a wireless communications technology, collaborating to forward the sensorial data hop-by-hop from the source node to the sink nodes.

In this paper, we mainly 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 TDMA protocols can provide low latency in the convergecast of data from the nodes to the sink, while still providing low duty-cycles. The quick convergecast is usually achieved by building 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 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 reception and transmission slots, while staying asleep for the rest of the TDMA frame.

With the objective of guarantying the same sensor-to-sink packet delay bound for all the nodes in the network, communication of alarms to the sink can be made in just one frame. However the size of the single frame is desirably the lowest possible. In this paper, several TDMA scheduling algorithms are simulated with the objectives of achieving low single frame sizes and low latencies.

This paper presents the related work in Section 2. In Section 3, the Cascading Minimum Single Frame Size Problem is defined. Section 4 presents the simulation model, the set of slot allocation algorithms, and their results in terms of achieving low single frame sizes, and low node-to-sink slot distances. Section 5 presents the simulations results obtained for a pre-determined frame size, in terms of node-to-sink slot distances. And finally, Section 6 presents simulation results, in terms of the actually required non-single frame sizes, and the respective worst-case delays of the communication of alarms to the sink.

RELATED WORK

While not being a pure TDMA protocol, the Data-gathering MAC (D-MAC) protocol, presented in Lu et al. (2004), uses staggered synchronization so that a data packet received by a node at one level of the tree, is transmitted to the next level in the following time period (i.e., cascading the transmissions in the overall transmission period). The node is then allowed to sleep until the reception period for its level occurs again. D-MAC is still a CSMA/CA based protocol as 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. Support of several sinks in D-MAC is troublesome.

The use of TDMA for fast broadcast (the converse problem of convergecast) is a well-known subject, which has been studied in the context of multi-hop radio networks. Chlamtac & Kutten (1987) 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, for which the first slot is granted, and builds a spanning tree, such that each node has a slot number higher than its parent slot, but with the smallest possible value, in order to cascade the broadcast. Tree construction and slot assignment are 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 obtain a TDMA schedule that meets some slot allocation rules, and that achieves 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, presented in Rhee et al. (2006). 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. DRAND assures that nodes in a 2-hop neighborhood do not use the same slot, and it can operate with limited frame sizes. DRAND is also proved to have a message exchanging complexity of O(δ), where δ is the neighborhood of each node.

(Annamalai et al., 2003; Upadhyayula et al., 2003), present two centralized algorithms, CTCCAA and CCA, which were specially designed to achieve low latencies in the convergecast process, namely by the use of cascading. 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. The two algorithms differ in the way they establish the neighborhood of each node for the purpose of avoiding conflict schedules. However, they present the drawback of being centralized and thus not adaptive to the irregular propagation characteristics of the environment.

Kulkarni & Arumugam (2005) present SS-TDMA, which 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 quotient y between them gives an estimation of the number of nodes within the 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. Those values depend on the direction of the message, the y value, and the type of the grid, namely square and hexagonal. Although being a distributed algorithm, it needs a location service and topological knowledge about the networks, which limits its practical applicability. SS-TDMA follows a 2-hop neighborhood interference avoidance criterion, but it allocates different slots for the purposes of broadcast, and convergecast.

The problems of building routing trees, and minimizing convergecast latency in ZigBee networks, were studied by Pan & Tseng (2008). 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.

Bryan et al. (2007) present a centralized algorithm and two distributed algorithms (namely, the DSA-AGGR – Distributed CCH for Data Aggregation). All the three algorithms are 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 allocate a slot if the following expression results in a value higher than 0.25 for the *color_score*:

$$color\_score = \frac{2.ColoredOneHop + ColoredTwoHop}{2.NumberOneHops + NumberTwoHops} \qquad (1)$$

This means that this heuristic gives priority to the nodes that have a sufficient number of neighbors that are already colored. This heuristic is claimed to obtain low frame sizes. Interference avoidance is also based on a 2-hop criterion. However, convergence of this algorithm is not always guaranteed, as some *color_score* thresholds can't be attained in some topologies.

Gandham et al. (2008) present a TDMA scheme that allows the implementation of convergecast and the cascading of the timeslots, which is based on the reduction of the networks, namely tree networks, to multi-line networks. In a given linear branch each node knows its hop distance to the sink, and can schedule itself to be in one of three states: T for transmit, R for receive, and I for idle. These states are rotated in time and arranged in such a way that the packets flow in a propagation wave towards the sink. For more complex topologies, the networks can be transformed into multi-line networks, but the nodes need to acquire knowledge about the global topology in order to know their turns to transmit upstream. This scheme has a major drawback, as the complexity of the basic scheme increases significantly when it is progressively extended to deal with more realistic scenarios.

Lu & Krishnamachari (2007) present a set of joint routing and slot assignment algorithms that aim to achieve low latencies. However, the procedures have the drawbacks of being computational intensive, while the tree building process is centralized.

Finally, in Mao et al. (2007) 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.

THE CASCADING MINIMUM SINGLE FRAME SIZE PROBLEM

Some of the works mentioned above are concerned with achieving short frame sizes, and therefore high throughput. As critical alarm-driven WSN applications should only report sporadic events, they do not need to periodically transfer bulk data to the sink. Therefore, achieving high throughput is not a specific design requirement and consequently the problem addressed in this paper is not to obtain the smallest possible frame sizes (i.e. maximum slot reutilization). The latter depends basically on the maximum degree (i.e., maximum number of neighbors) of the networks. Moreover, small frames 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 nodes located at different levels of the network tree. In fact, the aim is to have similar delay bounds for the alarms transmitted by all the network nodes. This requirement can be accomplished by always transmitting the data in a single TDMA frame, whatever the location of the node. However, since different scheduling algorithms may lead to different single TDMA frame sizes, the objective is to find algorithms that lead to the smallest possible single TDMA frame size in such a way that it is able to accommodate transmissions from all the network nodes, spanning from the deepest leaf nodes to the sink node. We call this problem the Cascading Minimum Single Frame Size (CMSFS) problem.

Differently from some protocols and algorithms presented here and generally in the literature, in this paper, it is assumed that the slot assignment procedure takes place during the network setup phase, but only after the routing tree construction. The latter is assumed to make use of an energy-efficient contention MAC protocol like B-MAC, as presented by Polastre et al. (2004).

SCHEDULING ALGORITHMS AND RESULTS FOR THE MINIMUM SINGLE FRAME SIZE PROBLEM

Centralized slot allocation algorithms are potentially more optimal and more predictable in terms of convergence, but they require that the nodes communicate their local topology (e.g., their neighborhoods, parents, etc.) to the processing node (usually the sink node), which is a slow and communication intensive procedure. 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 centralized strategies referred in the literature, or those that build the tree simultaneously with the allocation process.

The following slot allocation strategies were firstly considered: depth-first (DF) and breadth-first (BF) (see, Cormen et al., 2000), RANDOM, DSA-AGGR, and SS-TDMA. Note that DF and BF can be easily distributed (see, Chalamtac & Kutten, 1987, for the case of distributed DF). In this paper, DF and BF are implemented in such a way that their distributed behavior is emulated with maximum possible fidelity, albeit without implementing the particular protocol details. The same design option was also adopted for all the remaining allocation algorithms. All of the implemented strategies perform greedy cascading slot allocation by each node.

The Degree Heuristic, presented in West (2001), which allocates the nodes ordered by the size of their neighborhoods, the similar Minimum Neighborhood First (MNF), and the Progressive Minimum Neighborhood First (PMNF) of Ramanathan (1999), were not considered because they are centralized, and also because their performance is expected to be similar to DSA-AGGR of Bryan et al. (2007).

The RANDOM strategy consists in selecting randomly and allocating any node whose parent node has already allocated 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, y was set to 2, meaning that the interference range was twice the communication range. In this way, if a node receives a message from its Northern closest neighbor (its parent), it allocates a slot number equal to its neighbor's minus one; if it receives a message from its Western closest neighbor, it allocates 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, also achieving spatial reutilization of the slots, while implementing a 2-hop neighborhood interference avoidance criterion. However, the SS-TDMA allocation assumes that the slots are unidirectional, and separates the slots that are destined to convergecast, from the slots that are destined to broadcast.

The slots that were allocated by SS-TDMA cannot be made bi-directional, or otherwise the 2-hop interference avoidance criterion is not valid anymore.

Simulations were carried out considering a 100-node square grid physical topology, where the sink node was placed at the upper-left 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 stay closer 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.

A simple free space propagation model was used, with the path loss exponent set to 2, and radio propagation irregularity was not considered (see, Rappaport, 2002) for the sake of simplicity. This model was selected because it was found to be the most common in the related literature.

The dimension of the grid square edges was set to the approximate value of the communication range. The interference range was set to twice the communication range. Therefore, each node had a maximum of twelve 2-hop neighbors. The interference graph was built based on a 2-hop neighborhood criterion, as this is also a customary assumption in the related publications. Since each slot was considered bi-directional, the links that were considered as potentially interfering with the parent-child communication were all the links established by the 2-hop neighbors of both the parent node and of the child node, using the same timeslot.

The dimension of each slot was configured to offer three transmission opportunities, in order to maximize the probability that successful packet transmission is still achieved in a single TDMA frame.

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, 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} topologies. For each of these 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 the branches in different sequences. These observations show that the number of possible slot schedules is a very huge number, being impossible to find the optimal TDMA single frame size based on an exhaustive search strategy. 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 considered to be sufficient to assess how the different slot allocation algorithms behave. The histograms of the frequencies for the respective TDMA frame sizes were then built. One reason that supports the decision of making only 10,000,000 simulation runs is that the obtained histograms are quite smooth in shape, resembling normal distributions. Probably even a smaller number of simulation runs would also be sufficient to obtain mean values with high accuracy, as high stability of the results was observed for different sets of simulation runs, but the problem is that some slot allocations, with slot counts approaching the edges of the distributions, would not appear. Since the RANDOM strategy can descend the tree in a much larger number of ways than the other disciplines, including, for instance, the breadth-first and the depth-first allocations as special cases, the number of simulations for this algorithm was raised to 100,000,000.

Figure 1 shows the histograms for the single frame size, obtained for the first set of slot allocation algorithms.

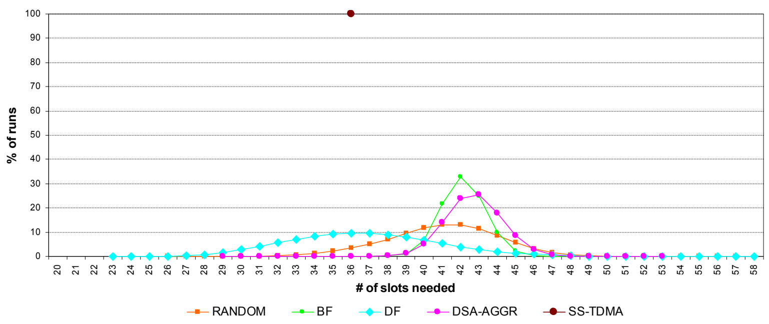

Figure 1. Single TDMA frame size histograms for the first set of slot allocation algorithms. The histograms are based on 10,000,000 simulation runs for each algorithm, with the exception of the 100,000,000 for the RANDOM algorithm.

Those 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 the five first rows of 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.

Table 1

Single frame size statistics for each allocation algorithm.

Algorithm	*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

It is interesting to note that SS-TDMA achieves always the same number of slots, i.e., 36, whatever the logical topology. The reason for this behavior is simple: the distances (in terms of slot numbers) between a given node and 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 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 in the allocation of the longest branch first are not generally sufficient to color all the other network nodes, and that in most of the times there is a need for some extra slots: for instance, DF achieved a minimum of 23 slots, while the longest path had always a length of 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, independently of their lengths. Therefore, in the allocation of the longest branches, the respective nodes find more slots that were already spent in previous slot assignments. Consequently, the slot distances between a parent node and its children are generally longer in those branches, 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 slots that were already assigned (sometimes to nodes placed deeper in the tree), resulting in larger frame sizes.

The remarks made above suggest that a longest-path-first strategy will lead to smaller TDMA frame sizes. In order to confirm this hypothesis, the next set of simulations considered a slot allocation algorithm that descends the tree in a longest-path-first scheme, 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). Longest-path first has been also used in the past in some similar contexts, namely for the wavelength division multiplexing (WDM) problem of high bandwidth optical WANs (see, for instance, Chlamtac et al., 1992).

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, whatever their positions in the network, but always beginning in a node whose parent was already assigned a slot. When the paths have equal lengths, ties are broken giving priority to the paths that are situated deeper in 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 previously described DF-LPF.

The other strategy (CENT-LDF, centralized largest-distances-first) is similar to CENT-LDF, but 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 as for CENT-LPF. In fact, largest-distance-first is not a novel concept, since sometimes - though rarely -, it appears in the literature.

A distributed version of CENT-LDF was developed, which is designated by DIST-LDF, whose pseudo-code is listed in Fig. 2, where Δ*t* represents the expected time needed to allocate a slot and *c* is a configurable constant value.

```
At any node that has already a slot:
    If there is more than one child:
        For the child with the longest path to a leaf:
            Allocate a slot to it, within the time interval $\Delta t$;
        For each other child $i$, not in the longest path:
            Allocate a slot, within the time interval = $c$ * (size-longest-
            path - size-path($i$)) * $\Delta t$;
    If there is only one child:
        Allocate a slot to it, within the time interval $\Delta t$;
```

(Note: if there are two or more children with the same path lengths, they are scheduled randomly, with a separation interval of Δt between them)

Figure 2. 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 firstly (and immediately) initiates the distributed allocation of the branch that presents the longest path, while the other branches have to wait an amount of time before also starting to allocate slots for their nodes. The respective amounts of time are proportional to the difference between 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. Therefore, nodes that are in branches that feature distances that are more distant from the sink are allocated first than nodes that are in branches that feature distances that are closer to the sink. Intuitively, in order for the allocations of the larger distance branches not to be disturbed by the allocations of shorter branches, the former have to be scheduled sufficiently in advance. It is worth to note that the DIST-LDF code is extraordinarily simple and easy to implement in a real distributed environment.

Fig. 3 presents the single frame size histograms, showing that there is a systematic improvement on the single frame sizes, when DF-LPF, CENT-LPF, CENT-LDF, and DIST-LDF (whose results are the same as those of CENT-LDF) are successively considered. Average values for the single frame size were respectively 27.9, 25.6, 24.9, and 24.9, for these four slot allocation algorithms, compared with the 36.7 of blind DF, as it can be seen in the last rows of Table 1. The LDF algorithms also produce the smallest range of values, among all the considered slot allocation algorithms. CENT-LDF, and DIST-LDF, present single frame sizes that range from 21 to 33 slots, while blind depth-first (DF) presented a much broader range, from 23 to 58 slots.

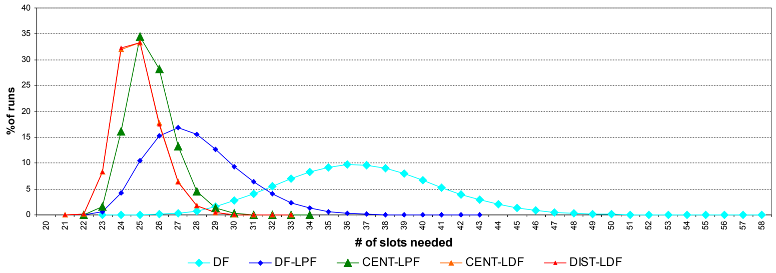

Figure 3. Single TDMA frame size histograms for the second set of slot allocation algorithms, showing improvements over simple blind depth-first (DF).

Fig. 4 and 5 are useful to understand the different behaviors of the BF and LDF disciplines, with respect to obtaining low single frame sizes. They show the allocations made by BF and DIST-LDF respectively, for some generated topology. In these figures, only a subset of 36 nodes of a total of 100 nodes in the network is shown. To the left of each node there is a pair of numbers: the first represents the order of the node in the resulting overall allocation sequence of the network, while the second represents the slot that was assigned to it.

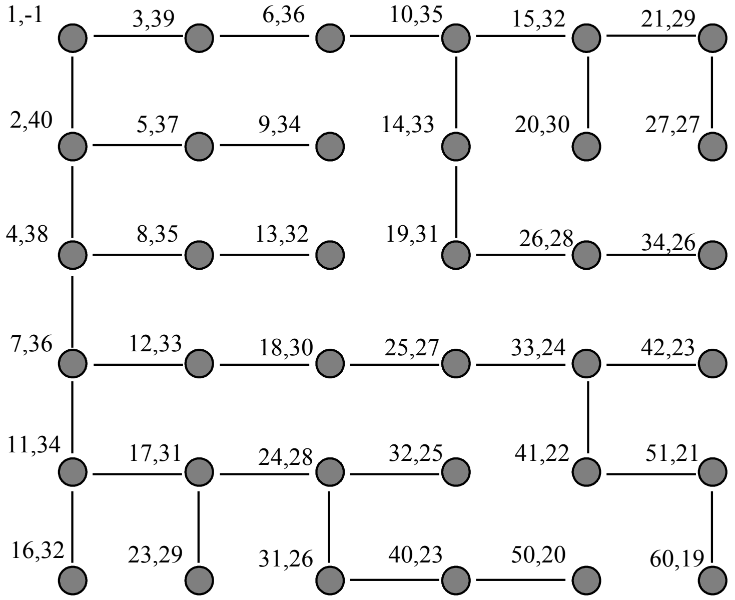

Figure 4. Example of one BF strategy allocation. BF generates larger single frame sizes.

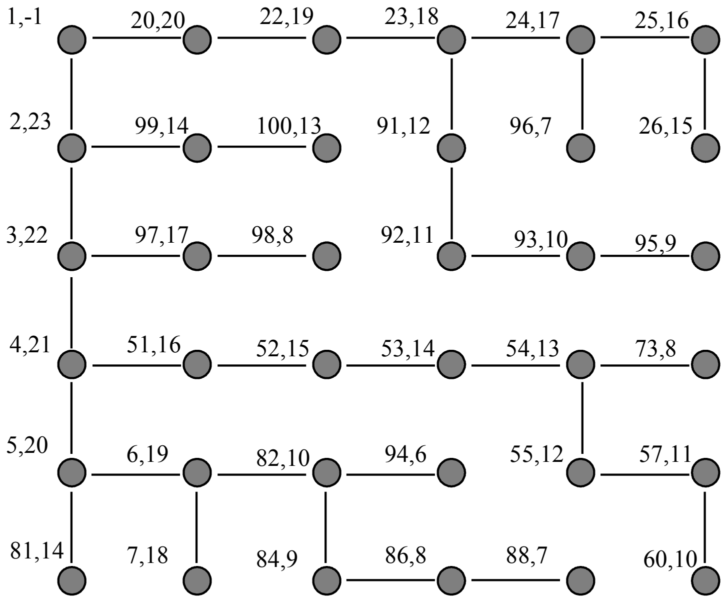

Figure 5. Example of one DIST-LDF strategy allocation. DIST-LDF generates smaller single frame sizes.

What is clear from these two figures is that BF allocates the consecutive nodes of any branch with slots that have always leaps between them, because it considers the allocation in a breadth manner, assigning slots at the sides of any branch, which cannot be reused by the nodes in the branch. On the other hand, DIST-LDF can allocate the largest distance and longest branches without leaps in the slot numbers assigned to consecutive nodes. Consider the first path (therefore the longest path in the network) that was allocated by the DIST-LDF algorithm (see Fig. 5). When the allocation process arrives to the second node of the lowest row, the slot distance to the sink is only 6 slots (23-18+1), while for BF (see Fig.4) the distance is already equal to 12 (40-29+1). The same phenomenon occurs in the second path allocated by DIST-LDF, namely the branch situated at the topside of the network. These different behaviors explain the better performance presented by DIST-LDF over BF, with respect to obtaining smaller single TDMA frames.

Regarding again the DIST-LDF code of Fig. 2, constant *c* was shown to generate the same results as the centralized algorithm when it takes values equal or greater than 5. For lower values, DIST-LDF performance degraded progressively into higher values for the TDMA frame size. In this case, some shorter distance branches do not wait for a sufficiently large time interval, and begin to allocate slots simultaneously with the longer distance branches. These concurrent actions are not separated enough in space and therefore the shorter distance branches can allocate slots that can't be used by longer distance branches, resulting in larger TDMA frames. In other words, longer distance branches are not scheduled sufficiently in advance. On the other hand, if the value of *c* is increased much beyond 5, the behavior of the DIST-LDF algorithm is not improved further, because the allocation of longer distance branches is already being performed at such distances that they cannot be affected by the allocations of shorter distance branches. Optimization of constant *c* for specific networks is, however, a subject for future work.

The slot distances from the nodes to the sink were also measured, and they are depicted in Fig. 6. CENT-LPF, CENT-LDF, and DIST-LDF achieved the lowest average slot distances (and also the lowest maximum slot distances), respectively with the average values of 14.8, 14.7, and 14.7 slots, against 20.4 for RANDOM and BF, 18.7 for DF, 20.5 for DSA-AGGR, 18.2 for SS-TDMA, and 15.6 for DF-LPF, meaning that they can also attain the lowest delays in the communication to the sink. DIST-LDF still presents the best performance.

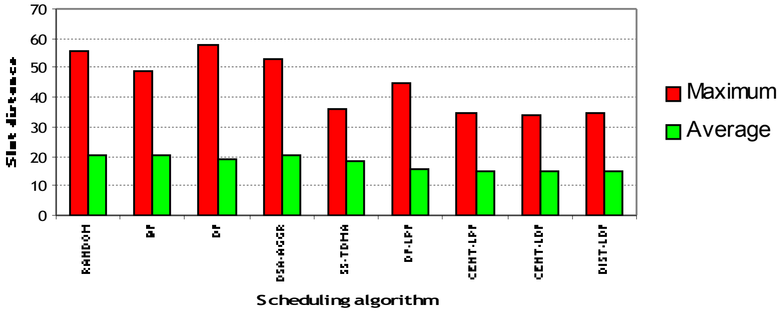

Figure 6. Maximum and average slot distances to the sink for the single frame scenario.

DIST-LDF presents the best results among the considered slot allocation strategies, but it requires each node to know the length of the branches that are rooted at each of its children. This may represent a significant disadvantage, since this information has to be propagated in the network tree, from the leaves to the sink, after the tree construction process. Such procedure may result in a significant overhead. Note, however, that the complexity of DIST-LDF in terms of the number of visited nodes is intuitively smaller than that of DF, since DIST-LDF does not need to backtrack in the tree structure, when it completely allocates a sub-tree. Therefore, it is also expected that DIST-LDF can achieve smaller execution times than DF. However, these two last observations need to be confirmed in future work.

SIMULATIONS RESULTS FOR A FIXED PRE-DETERMINED FRAME SIZE

Simulations have also been carried out to investigate the behavior of the scheduling algorithms, when the frame size is set to a fixed value, allowing that the sensor-to-sink communication can be completed in more than one frame.

Fig. 7 shows the results of the nine different scheduling algorithms, for a fixed frame size of 30 slots, and for a grid network of 100 nodes. This frame size of 30 slots was found to be enough to avoid having nodes that could not be assigned a slot in any of the simulation runs. All simulation procedures and parameters were kept the same as in the previous section.

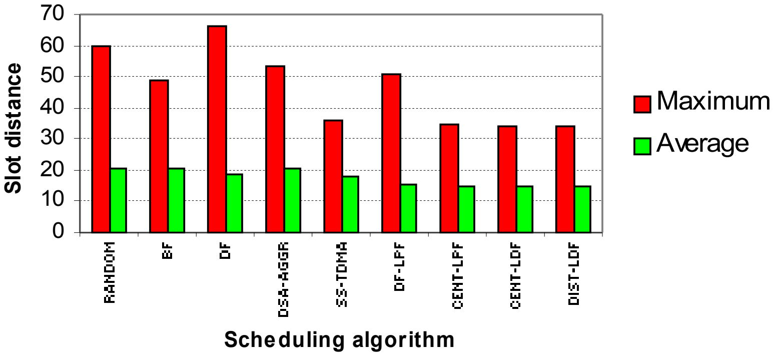

Figure 7. Maximum and average slot distances for a fixed frame size of 30 slots.

These results are very similar to those that were obtained for the single frame scenario: the proposed disciplines DF-LDF, CENT-LPF, and CENT-LDF, and DIST-LDF, featured a successive decrease on the slot distances. The explanation for this behavior follows the overall rationale that stands behind their design: when the smaller branches are allocated first, the longer branches will feature longer slot distances due to the increased difficulty of conflict-free cascaded allocation. Therefore, it is more advantageous to perform an allocation that results in larger slot distances on smaller branches, than to place larger slot distances in longer braches.

This is achieved by allocating longer branches first and results in a more uniform branch delay bound distribution.

Similar results were also obtained with larger networks of 400 nodes. These results suggest that CENT-LPF, CENT-LDF, and DIST-LDF disciplines can also attain lower communication delays when more than one fixed sized frame is required to transport data from leaf nodes to the sink.

SIMULATION RESULTS FOR THE ACTUALLY REQUIRED FRAME SIZES IN A NON-SINGLE FRAME SCENARIO

Two questions can now be made: how do the different scheduling algorithms behave with respect to the minimum non-single frame sizes that they actually require? And how they behave with respect to slot distances that they achieve, when they use the frame lengths that they actually require?

In order to answer to these questions, a third set of simulations was carried out. The algorithm that was used to determine the required minimum frame sizes was the following: since each node in the grid has a maximum of 12 neighbors, the initial frame size is set to 9 (this number resulted to be adequate, since the obtained simulation runs always required 10 or more slots.). When a node tries to allocate a slot, it firstly tries to find a slot that is right next to its parent’s slot, otherwise it tries each successive slot in the frame (eventually returning to the beginning of the frame), until a conflict-free slot is found or until it reaches its parent’s slot. If no conflict-free slot is found, the number of slots is increased by one and the node tries to allocate the new slot. It may happen that a node will find the new slot already occupied by one of its neighbors in which case it will have to increase the frame size again and retry. In fact, the frame size is kept as a local variable, since in a distributed algorithm it is not feasible to communicate local frame size changes to distant nodes. At the end of the slot allocation process, the frame size is equal to the maximum slot number that was needed by all the nodes of the network.

Fig. 8 shows the histograms of the required frame sizes for random topologies and allocations. And Fig. 9 shows the respective worst-case delays in terms of slot-distances to the sink. The worst-case delays take into account a possible delay bias of one frame, for the case of a packet being generated just after the respective slot has elapsed.

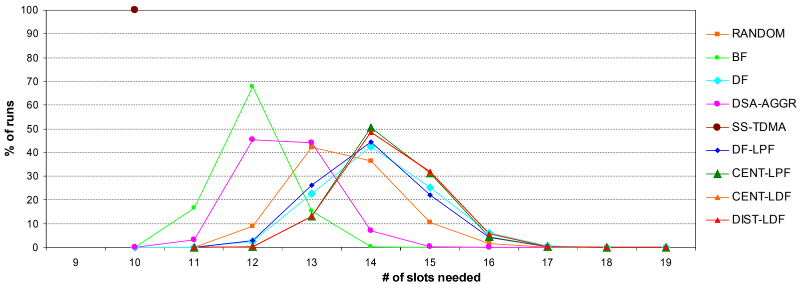

Figure 8. Histograms of the minimum non-single TDMA frame sizes actually required by the nine allocation algorithms.

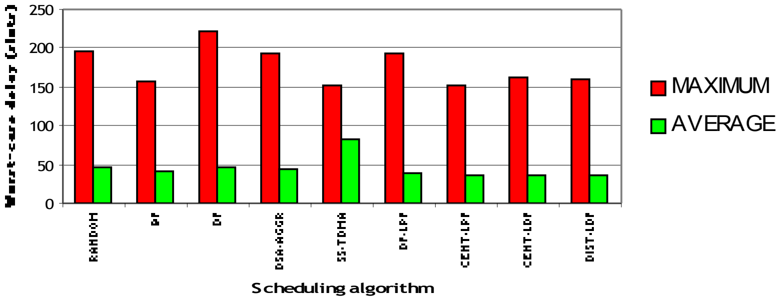

Figure 9. Maximum and average worst-case delays in terms of slot distance, for the non-single frame scenario.

Similar results were also obtained for larger networks of 400 nodes, but they are not shown for lack of space.

BF is known to be an efficient discipline with respect of requiring small frames, which is confirmed by the simulation results, which take into account cascading of TDMA slots.

BF achieved the lowest average value (12.0 slots) for the frame size, while the disciplines that descend the tree in-depth (DF, DF-LPF, CENT-LPF, CENT-LDF, and DIST-LDF) presented the highest values, being close to each other, (14.1, 14.0, 14.3, 14.3, and 14.3 respectively). DSA-AGGR (12.6) is closer to BF than to DF. The RANDOM discipline (13.5) is closer to DF than to BF. SS-TDMA presented always a fixed value of 10 slots but, as already noted, its slots are not bi-directional, separating broadcast slots from convergecast slots. Therefore, it requires fewer slots. However, SS-TDMA performance, in terms of worst-case delays, does not benefit from its smaller frame sizes.

It is worth to point out that BF behaves quite differently in the slot allocation process when compared with the disciplines that descend the tree in depth, requiring fewer slots. The reason for its lower frame sizes rests in the fact that when BF allocates a slot for a given node, it finds only about one half of its interfering neighbors with a slot already assigned. These are all the neighbors located at the upper levels of the tree and some of the neighbors located at the same level. In contrast with this, in the disciplines that perform the allocation in-depth, when a node attempts to allocate its slot, it may find that almost all of its neighbors have already allocated one slot. These neighbors can be located in any adjacent and previously allocated branch, since after descending some branch in-depth, these disciplines can backtrack to any other branches, including of course those that have nodes in the neighborhood of the node that is trying to allocate a slot.

The same reasoning can also be used to explain the poorer results of RANDOM and DSA-AGGR when compared with BF. Note that the RANDOM and the DSA-AGGR disciplines may sometimes descend the tree in a more in-depth fashion.

Comparing these results with those of Section 4, it is interesting to note that the scheduling algorithms behave in opposite ways when cascading allocation is employed. BF is the best discipline to achieve minimum non-single frame sizes in comparison with in-depth disciplines, while the opposite happens regarding the minimum single frame sizes.

Finally, regarding worst-case delays, it should be noted that Fig. 9 confirms the tendency of the longest-paths first and largest-distances first disciplines to present the lowest values, even if they need slightly larger frame sizes.

CONCLUSIONS

In most alarm-driven WSN applications, traffic can be characterized as very sporadic, but the generation of an alarm report demands an immediate response to the event. Low latencies and low duty-cycles can be simultaneously accomplished by using TDMA protocols. Cascading of slots may result in low latencies, while TDMA protocols can achieve low duty-cycles because each node only needs to be active in its own slots. Since this kind of applications does not need high throughputs, and it is also desirable to have the same delay bound for all the nodes in the network, we defined and investigated 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 (a problem that we have designated the Cascading Minimum Single Frame Size – CMSFS – problem).

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 would 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. This discipline is surprisingly simple, and easy to implement in real distributed slot allocation scenarios.

Simulations were also carried out to assess the behavior of the disciplines with respect to the slot distances and worst-case delays of the communication to the sink, for three scenarios: single frame sizes, non-single frame with a fixed frame size, and the non-single frame sizes that the algorithms actually require. In all these scenarios, the depth-first with the longest-path first, the longest-paths first and the largest-distances first disciplines, and its distributed version, presented successively lower values. This happened even if the in-depth disciplines required slightly larger non-single frame sizes than breadth-first. A comprehensive explanation for these last results was also provided in the paper.

As a main conclusion, it can be said that the largest-distances first discipline, and namely its distributed counterpart (DIST-LDF), can be a promising slot allocation strategy in order to obtain low single frame sizes and simultaneously obtain low delays in the communication to the sink, in a convergecast scenario.

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

Annamalai, V., Gupta, S.K.S., & Schwiebert, L. (2003). On Tree-Based Convergecasting in Wireless Sensor Network. *Proceedings of the IEEE Wireless Communications and Networking (WCNC 2003)*, New Orleans, LA, USA.

Bryan, K.L., Ren, T., DiPippo, L, Henry, T., & Fay-Wolfe, V. (2007). *Towards Optimal TDMA Frame Size in Wireless Sensor Networks* (Technical Report, TR-xxx). University of Rhode Island.

Chlamtac, I., & Kutten, S. (1987). Tree-based Broadcasting in Multihop Radio Networks. *IEEE Transactions on Computers, Volume* C-36, No. 10.

Chlamtac, I., Ganz, A., & Karmi, G. (1992). Lightpath communications: an approach to high bandwidth optical WAN's. *IEEE Transactions on Communications, Vol. 40, Issue 7*, pp. 1171-1182.

Cormen, T.H., Leiserson, C.E., & Rivest, R.L. (2000). *Introduction to Algorithms*. The MIT Press.

Gandham, S., Zhang, Y., & Huang, Q. (2008). Distributed Time-optimal Scheduling for Convergecast in Wireless Sensor Networks. *Computer Networks, Vol. 52, Issue 3*, pp. 610-629, Elsevier.

Kulkarni, S.S., & Arumugam, M.(U.) (2005). SS-TDMA: A Self-Stabilizing MAC for Sensor Networks. *Sensor Network Operations*, IEEE Press.

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

Lu, G., & Krishnamachari, B. (2007). Minimum latency joint scheduling and routing in wireless sensor networks. *Ad Hoc Networks, Vol. 5, Issue 6*, pp. 832-843, Elsevier.

Mao, J., Wu, Z., & Wu, X. (2007). A TDMA scheduling scheme for many-to-one communications in wireless sensor networks. *Computer Communications, Vol. 30, Issue 4*, pp.

863-872, Elsevier.

Pan, M.S., & Tseng, Y.-C. (2008). Quick convergecast in ZigBee beacon-enabled tree-based wireless sensor networks. *Computer Communications, Vol. 31, Issue 5*, pp. 999-1011, Elsevier.

Polastre, J., Hill, J., & Culler, D. (2004). Versatile Low Power Media Access for Wireless Sensor Networks. *Proceedings of the 2nd ACM SenSys Conference*, pp. 95-107, Baltimore, MD, USA.

Ramanathan, S. (1999). A unified Framework and Algorithm for Channel Assignment in Wireless Ad Hoc Networks. *Wireless Networks, Vol. 5, no. 2*, pp. 81-94.

Rappaport, T. (2002). *Wireless Communications: Principles and Practice*. 2nd Edition, Prentice Hall.

Rhee, I. Warrier, A., Min, J., & Xu, L. (2006). DRAND: Distributed Randomized TDMA Scheduling for Wireless Ad-hoc Networks. *Proceedings the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'2006)*, Florence, Italy.

Upadhyayula, S., Annamalai, V., & Gupta, S.K.S. (2003). A low-latency and energy-efficient algorithm for convergecast in wireless sensor networks. *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM '03)*, San Francisco, CA, USA.

West, D.B. (2001). *Introduction to Graph Theory*. 2nd edition, Prentice Hall.