Efficient Data Aggregation Scheduling in Wireless Sensor Networks with Multi-Channel Links

Miloud Bagaa
Dep. of Theo and,
Comp Eng, CERIST,
Algiers, Algeria
bagaa@mail.cerist.dz

Mohamed Younis
Dep. of Comp Sc.
& Elect Eng, Univ. of
Maryland Baltimore County
younis@cs.umbc.edu

Nadjib Badache
CERIST, Algiers, Algeria
badache@mail.cerist.dz

ABSTRACT

In-network data aggregation is often pursued to remove redundancy and correlate the data en-route to the base-station in order to save energy in wireless sensor networks (WSNs). In this paper, we present a novel cross-layer approach for reducing the latency in disseminating aggregated data to the base-station over multi-frequency radio links. Our approach forms the aggregation tree with the objective of increasing the simultaneity of transmissions and reducing buffering delay. Aggregation nodes are picked and time-slots are allocated to the individual sensors so that the most number of ready nodes can transmit their data without delay. Colliding transmissions are avoided by the use of different radio channels. Our approach is validated through simulation and is shown to outperform previously published schemes.

Categories and Subject Descriptors

C.2.1 [**Computer Communication Networks**]: Network Architecture and Design—*Wireless communication, Network topology*; F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*Sequencing and scheduling*

Keywords

wireless sensor network, data aggregation, multi-channels, media access scheduling

1. INTRODUCTION

Wireless sensor networks (WSNs) are composed of a large number of miniaturized and low-cost wireless sensor nodes, that can gather data about some phenomena in the environment. Each sensor node can collaborate with other sensors to route the sensed data to a processing center, which is also called the base-station or sink node. However, the sensor nodes have limited resources in term of battery power, computational capacity, and communication bandwidth. To save these resources, aggregation of the data collected by the individual sensors en-route to the base-station is one of the popular optimization strategies. Data aggregation opts to eliminate redundant packet transmissions by filtering repeated and unnecessary data readings and thus cut on the energy used in communication.

WSNs have numerous applications such as, early detection of forest fire [4] and security surveillance [3]. In these applications, it is very critical to deliver alarms about serious events in a timely manner so that an appropriate action can be taken in response. Therefore, the network design should strive to minimize the delay in disseminating the data to the base-station. One of the most effective schemes is to limit medium access latency by adopting reservation based protocols such as *TDMA* (Time Division Multiple Access). In-network data aggregation helps in limiting medium access contention and reduces the number of transmitted packets, and thus can help in finding low-latency medium access schedule.

Recently cross-layer methodologies have been deemed advantageous and a number of techniques have emerged to combine the formation of the aggregation tree with medium access scheduling [2] and [5]. These solutions assume that the network nodes use only one channel to communicate. However, the use of multi-channel radio transceivers in *WSN* has recently become popular. Exploiting the availability of multiple channels would allow increased simultaneous transmissions, i.e., time slots reuse, and would thus reduce the data latency. In this paper, we pursue a novel cross-layer approach for establishing data aggregation over multi-channel links (*DAS-MC*) that minimizes data delivery latency. *DAS-MC* considers setups in which each node can transmit and receive on multiple radio frequencies. Unlike published solutions, *DAS-MC* intertwines the tree construction and transmission scheduling. *DAS-MC* does not purse breadth-first ordering. When a node that is L-hop away from the sink (i.e., in level L) becomes ready to schedule, it selects its parent from nodes in level $L-1$, L, or $L+1$ and its time slot as the earliest one in all the channels. This relaxed approach, which has not been proposed before, lowers the latency and increases the network throughput.

The rest of this paper is organized as follows. Related work is summarized in section 2. The problem formulation and the basic idea of our approach are presented in section 3. The approach details are presented in section 4. The simulation results are discussed in section 5. Finally, the paper is concluded in section 6.

2. RELATED WORK

The existing solutions aim to maximize the network throughput by gathering the data from all network nodes in the

MSWiM'13, November 3–8, 2013, Barcelona, Spain.

http://dx.doi.org/10.1145/2507924.2507995.

least time. To observe data freshness requirements, the slot allocated to a parent would be greater than all the ones of its children. Chen et al. [2] have proved that the minimum delay data aggregation scheduling problem is *NP-hard*, and proposed an iterative scheduling procedure applied on the formed tree. Huang et al. [5] proposed a centralized solution, called *NCA*. *NCA* uses the Connected Dominating Set (*CDS*) as an aggregation tree. Malhotra et al. [8] observed that the data path from the nodes to the sink in the *CDS* based aggregation tree is not necessarily optimal and proposed an algorithm, which uses a *Balanced Shortest Path Tree* instead of a *CDS* as a data aggregation structure. Bagaa et al. proposed a centralized algorithm in [1] in which no information about candidate parents or children are provided to the scheduling algorithm. The tree construction and the scheduling are performed concurrently.

However, the proposed solutions in [2, 5, 8, 1] do not support multi-channel links when scheduling network nodes. For this reason the solutions [6], [7], [9] and [10] have been proposed to exploit the capabilities of the new transceiver. The network nodes in [10] are organized into clusters. The cluster head aggregates the data and routes the cell report to the base-station over an inter-cluster-head tree. To increase the network throughput, each branch of the tree is assigned a different channel. Receiver-Based Channel Assignment (*RBCA*) has been proposed in [6]. In *RBCA*, the children of the same parent use the same channel (i.e., the parent's channel) to transmit their data. A Tree-Based Multichannel Protocol (*TMCP*) has been proposed in [9]. The aggregation tree in *TMCP* is divided into a set of subtrees, such that the root of each of them is a neighbor of the sink. In order to minimize the interference in the subtrees, *TMCP* assigns a different channel for each subtree. Meanwhile, a Joint Frequency Time Slot Scheduling (*JFTSS*) has been proposed in [7]. The links of the tree are sorted in *JFTSS*, such that the link which conflicts with most number of other links has high priority for channel reassignment. However, these solutions de-couple the tree construction from the nodes scheduling. Moreover, in the approaches of [9], [7] and [6], the parent's transmission slot can be earlier than the child' one, which leads to aggregating the data of different periods together. Thus, the data freshness is not ensured.

3. PROBLEM DEFINITION AND SOLUTION STRATEGY

Devising a non-conflicting medium access schedule that achieves minimum delivery latency subject to flow constraints is a known NP-hard problem [2]. Therefore, heuristics are usually pursued. *DAS-MC* opts to reduce time latency by increasing the time slot reuse and by selecting for each node the smallest time slot in all the channels which ensures the aggregation freshness. *DAS-MC* intertwines the tree construction and nodes scheduling. Therefore, at each moment in *DAS-MC* execution, each node in the network can have two kinds of neighbors: (i) a set of nodes which has already designated its parent and set its time slot (i.e., scheduled node); (ii) set of nodes which are not yet scheduled. For each channel i the neighbors $\gamma(u)$ of a node u can be grouped into three disjoint subsets according to a time slot τ. The subset $\delta_i^\tau(u)$ includes nodes that, (a) are not selected as parent at time slot τ on channel i, and (b) cannot hear at this time slot using channel i due to collision. Each node v in $\delta_i^\tau(u)$ has at least one node in $\gamma(v)$ which has already designated its parent and set its time slot to τ and its channel to i (i.e., scheduled node); The subset $\varrho_i^\tau(u)$ has nodes which are selected as parents by other scheduled nodes at time slot τ and channel i. The neighbors of nodes in this subset are prevented from using τ on channel i due to collision. The subset $\varphi_i^\tau(u)$ has the nodes which can be parents of node u at time slot τ on channel i and can be formally defined as:

$$\varphi_i^\tau(u) = \gamma(u) - (\delta_i^\tau(u) \cup \xi^\tau(u) \bigcup_{j=1}^{k} \varrho_j^\tau(u)) \tag{1}$$

such that $\xi^\tau(u)$ is the set of u's neighbors which are (i) scheduled and (ii) their time slot lower or equal to τ. We remove $\xi^\tau(u)$ from $\varphi_i^\tau(u)$ to ensure data freshness and prevent the creation of cycles. If $\varrho_i^\tau(u) \neq \emptyset$ or $\varphi_i^\tau(u) = \emptyset$, node u should select a time slot that is higher than τ in order to avoid introducing collision.

Each node u in *DAS-MC* should use only one channel CH_u and one time slot TS_u to transmit its data to the parent node $\rho(u)$. CH_u and TS_u should be selected such that the time latency is reduced. Therefore, TS_u should be selected as the smallest time slot in all the channels, whereas CH_u should the channel which does not cause the collisions and also allows more time slot reuse. *DAS-MC* aims to capture this trade-off by assigning for each node u the time slot TS_u and channel CH_u which reduce the time latency and increase the potential of the time slot reuse.

While all nodes in $\gamma(u)$ cannot receive another packet when u transmits at time slot TS_u on channel CH_u, a node $v \in (\gamma(u) - \{\rho(u)\})$ can transmit to a node $w \in \gamma(v)$ at TS_u on channel CH_u as long as $w \notin \gamma(u)$ and $\rho(u) \notin \gamma(v)$. Moreover, if v uses a channel different than CH_u, it can transmit to any node in $w \in \gamma(v) - \{\rho(u)\}$ at TS_u. *DAS-MC* leverages this observation to boost the simultaneity of transmissions, and consequently minimize the data delivery latency, by increasing the chance to find a u's parent $\rho(u)$ in which u can use a smallest time slot. To select the smallest time slot for a node u, we should increase the size of $\varphi_i^\tau(u)$ for each channel i as much as possible. To achieve this objective, *DAS-MC* intertwines the tree construction and data aggregation scheduling. In addition, *DAS-MC* allows a node u in level L, where L is the shortest distance between u and the sink, to select its parent from its neighbors in (1) level $L-1$ (upper-level), (2) level L (same-level), and (3) level $L+1$ (lower-level) providing that does not create cycle. Favoring parents that have the fewest number of neighbors in and not yet scheduled enables increased time slots reuse.

4. MULTI-CHANNEL DATA AGGREGATION SCHEDULING

To reduce the time latency three fundamental issues are to be addressed; (1) what the transmission order constraints among nodes are, (2) how channels are assigned to links, and (3) how parents and children are associated on the aggregation tree so that time slot reuse is maximized. The balance of this section opts to explain the operation of *DAS-MC* and how it tackles these issues. In addition to CH_u, TS_u, $\rho(u)$, $\gamma(u)$, δ_i^τ, ϱ_i^τ, φ_i^τ defined earlier, the following notation will be used in the discussion:

- (u, v): Link between the nodes u and v.
- SC_u: Set of nodes in $\gamma(u)$ which are scheduled, i.e., the nodes which already select their parents and set their time slot.

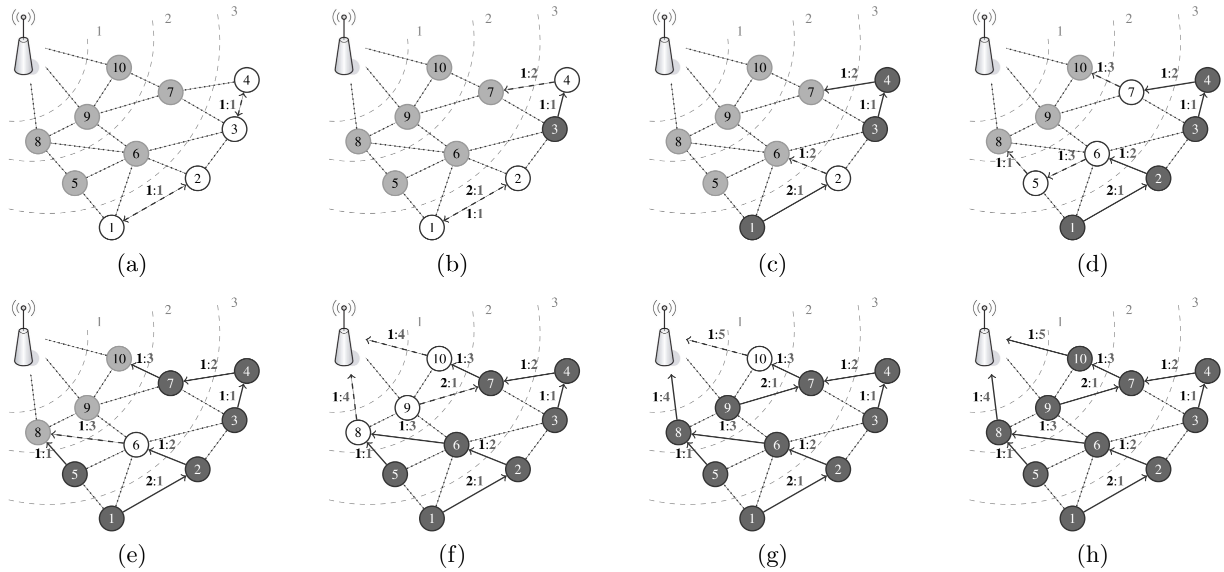


Figure 1: Detailed illustrative example to show how DAS-MC is applied

- $\overline{SC_u}$: Set of nodes in $\gamma(u)$ which are unscheduled i.e., the nodes in $\gamma(u)$ which have not yet selected their parents. Formally, $\overline{SC_u} = \gamma(u) - SC_u$.
- T_i^j: The use of time slot j on channel i.
- η_i: Node whose identifier is i. η_0 is the base station.
- $\psi_i(u)$: The parent of node u on channel i.
- $TC(u)$: The maximum time slot of u's children, initially is set to zero.

In order to facilitate the presentation of *DAS-MC*, Figure 1 will be referenced throughout the discussion. Figure 1 shows an example of *DAS-MC* execution. In this example, we consider that there is only two channels and the depth of the breadth-first tree is three (i.e., 3 levels). In the figure, the gray, white and black circles represent the unready, ready and scheduled nodes, respectively. A dashed arrow between two nodes a and b indicates that a has chosen b as its best parent when the arrow becomes solid it indicates that the link (a, b) is scheduled. The dotted lines represent the graph connectivity. The gray number besides the dashed and solid arrow (a, b) represents a's transmission slot. The black number besides the dashed and solid arrow (a, b) represents a's transmission channel.

4.1 Temporal ordering of transmissions

The nodes in *DAS-MC* are scheduled bottom-up, level by level in a breadth-first order. Each node in level L will not get scheduled until all its $L+1$ neighbors are assigned transmission slots. For example in Figure 1(a) η_1, η_2, and η_3, which are in level L, should be scheduled before η_5, which is in level $L-1$. The scheduling of nodes in a breadth-first order does not mean that the aggregated data is also routed on a breadth-first tree. In *DAS-MC* when a node in level L gets scheduled, it selects its parent from $L-1$, L or $L+1$ levels. For example in Figure 1(c) η_2, which is in level L, selects its parent η_6 from level $L-1$, whereas in Figure 1(f) η_9, which is in level L, selects its parent η_7 from level $L+1$.

4.2 Assigning parents and time slots

Let T_i^τ be the earliest time slot on channel i which ensures the data freshness and prevents message collision. Formally T_i^τ is defined as the earliest time slot (i) which is higher than all u's children and (ii) which fulfills that $\varrho_i^\tau(u) = \emptyset$ and $\varphi_i^\tau(u) \neq \emptyset$. In *DAS-MC* $\psi_i(u)$ is selected from $\varphi_i^\tau(u)$ as the node which has the fewest number of neighbors and not yet scheduled to encourage time slots reuse. That is if a node u selects $\psi_i(u)$ which has many unscheduled neighbors, the set ϱ_i^τ at many nodes will not be empty and then many nodes will be prevented from reusing T_i^τ.

A node in *DAS-MC* should use only one channel CH_u and one time slot TS_u to send the aggregated data to its parent $\rho(u)$. CH_u, TS_u and $\rho(u)$ should be selected such that (1) the messages collision is prevented, (2) data freshness is ensured, and (3) the time latency is reduced. To reduce the time latency, CH_u is set to be the channel i which offers the earliest time slot T_i^τ. If there is many channels which qualify, CH_u is picked to enable the selection of the parent with the fewest number of neighbors and not yet scheduled. If there is more than one channel which fulfill the above conditions, the channel with the smallest *ID* will be selected. Meanwhile, $\rho(u)$ and TS_u will be selected according to CH_u value. $\rho(u)$ and TS_u are set to $\psi_{CH_u}(u)$ and $T_{CH_u}^\tau$ values, respectively. In the selection of CH_u, *DAS-MC* gives the first priority to the smallest TS_u then the parent $\rho(u)$ with smallest number of unscheduled neighbors. If we give priority to the parent which has the fewest unscheduled neighbors many time slots never be used creating gaps, and thus the time slot reuse will diminish.

In the previous solutions, when a node u gets scheduled, it should select its parent from $\overline{SC_u}$ to prevent creation of cycles. Therefore, the gaps slot is never used and thus the time slot reuse will be significantly decreased. Unlike the published solutions, *DAS-MC* explores the gaps slots by allowing each node u to select its parent from scheduled nodes, i.e., $\{\rho(u)\} \cap SC(u) \neq \emptyset$. From equation 1 a node u in *DAS-MC* selects its parent w from $\varphi_i^\tau \cap SC_u$ if and only if $TS_u < TS_w$, thus the data freshness is ensured and creation of cycles are prevented. Finally, it is important to note that the parent selection criteria of *DAS-MC* allows the parent of a node in level L to be picked from level L, $L-1$ or $L+1$, and thus the aggregation is not restricted to a breath-first tree as on previously published work.

To illustrate, let us use node η_1 in the example in Figure 1(a). $\varrho_1^{\#1}(1) = \emptyset$, and $\varphi_1^{\#1}(1) = \{\eta_2, \eta_5, \eta_6\}$. The earliest available time slot for channel 1 that can be assigned to η_1 is #1. The number of nodes in $\overline{SC}$ of η_2, η_5 and η_6 are 3, 3 and 6, respectively. It is obvious that the choice which maximizes the time slot reuse is $\psi_1(1) = \eta_2$ or $\psi_1(1) = \eta_5$. As η_2 has lowest ID, $\psi_1(3) = \eta_2$. For the same reason in channel 2 the earliest available time slot that can be assigned to η_1 is #1 and its parent $\psi_2(1) = \eta_2$. As the two channels offer the same time slot and whose parents have the same number of unscheduled neighbors, η_1 selects the channel which has the smallest ID, which is 1. Thus, CH_1, TS_1 and $\rho(1)$ will be set to 1, #1 and η_2, respectively. In this case, η_1 which is in level L selects its parent $\rho(1)$ from the same level. Figure 1(f) shows an example to illustrate the selection of a parent of η_9 from scheduled nodes and level $L+1$. In the Figure $\varrho_1^{\#1}(9) = \{\eta_8\}$, which means that the time slot #1 with channel 1 is reserved by the children of η_8, and if η_9 uses this time slot on channel 1 a collision will occur at η_8. If node η_9 uses channel 1 with time slot 2 (resp., 3) a collision will occur at node η_6 (resp., η_8). $\varrho_1^{\#4}(9) = \emptyset$ and $\varphi_1^{\#4}(9) = \{\eta_0, \eta_8, \eta_1 0\}$, the earliest time slot that can be assigned to η_9 on channel 1 is #4 and its parent $\psi_1(9) = \eta_8$. In the same Figure, $\varrho_2^{\#1}(9) = \emptyset$, $\xi^{\#1}(9) = \emptyset$, $\delta_2^{\#1}(9) = \{\eta_6\}$ and $\varphi_2^{\#1}(9) = \gamma(9) - (\delta_2^{\#1}(9) \cup \xi^{\#1}(9) \bigcup_{j=1}^{2} \varrho_j^{\#1}(9)) = \{\eta_0, \eta_7, \eta_{10}\}$. As η_7 has the smallest number of unscheduled neighbors, the earliest time slot that can be assigned to η_9 on channel 2 is #1 and its parent $\psi_2(3) = \eta_7$. As channel 2 offers the earliest time slot, CH_9, TS_9 and $\rho(9)$ will be set to 2, #1 and η_7, respectively.

4.3 Scheduling the transmitting nodes

The parent selection step discussed above is node-centric in the sense that it is limited to node u when picking a parent, and assigning a time slot and channel. However, conflicts may exist among the individual nodes which are not neighbors on the same level L. To illustrate, consider the example in Figure 1(a). Based on the parent selection criteria discussed above, nodes η_1 and η_2 each one select the other as its parent at $T_1^{\#1}$. Nodes η_3 and η_4 are also each one select the other as its parent at $T_1^{\#1}$. If η_1, η_2, η_3 and η_4 are scheduled at $T_1^{\#1}$ a collision will be occurred. *DAS-MC* resolves such a conflict by considering every pair for ready nodes u and v in the level that is being scheduled. A conflict exists between nodes u and v if one of the following conditions is hold:

1. A collision: If $TS_u = TS_v$ and
   (a) $\rho(u) = \rho(v)$ or
   (b) $CH_u = CH_v$ and ($\rho(u) \in \gamma(v)$ or $\rho(v) \in \gamma(u)$).
2. Violation of data freshness or creation of cycle: If a ready node selects another one as its parent i.e., $\rho(u) = v$ or $\rho(v) = u$.

To resolve a collision, *DAS-MC* picks the one among the two nodes u and v whose parent collectively has the least number of unscheduled interfering nodes (i.e., smallest size of $\overline{SC}$) and reschedule its transmission in order to enable the most reuse. In case of a tie, the node, either u and v, that has the least number of unscheduled interfering nodes will keep the earliest slot in order to allow as many nodes as possible to be parent in this slot with channel CH_u, i.e., reduce the size of $\delta_{CH_u}^{TS_u}$. In the case both u and v, fulfill the above conditions, the node with the smallest ID will be scheduled. Meanwhile, the violation of data freshness or creation of cycles are resolved as follows: If $\rho(u) = v$ or $\rho(v) = u$ then the node which has the earliest time slot, say u, should be scheduled first. That is if node v sets u as its parent and v is scheduled first, node u should change its time slot to be later than TS_v to ensure the data freshness. If $TS_u = TS_v$, the conflict will be resolved using the same approach.

Coming back to the example in Figure 1(a), let us consider the four nodes η_1, η_2n η_3 and η_4. For η_1 and η_2, if η_1 is scheduled first, the set $\varrho_1^{\#1}$ of the two unscheduled interfering nodes of η_1's parent (i.e., $\overline{SC}_2 - \{\eta_1\}$), will not be empty and will be prevented from transmitting at $T_1^{\#1}$. Meanwhile, assigning slot #1 to η_2 makes η_1 locked and will affect also 2 unscheduled interfering nodes (i.e., $\overline{SC}_1 - \{\eta_2\}$). Moreover both nodes prevent the same number of nodes from serving as parents at time slot TS_u on channel CH_u (i.e., they make $\delta_i^{\tau}(u)$ of the same number of nodes not empty), so the node which has the smallest ID would be scheduled first. Thus, node η_1 should be scheduled before node η_2. As there is a conflict between nodes η_1, η_3 and η_4, one of them should be scheduled first. Node 4 prevents three nodes to use time slot #1 (i.e., $\overline{SC}_3 - \{\eta_4\}$), whereas assigning slot #1 to η_3 makes η_4 locked and will affect only one node. Therefore, nodes η_3 is to be allocated $T_1^{\#1}$ before η_2, η_4 and η_1 as depicted in Figure 1(b). Such a decision leads to higher time slot utilization, and hence minimum latency is likely to be achieved.

5. SIMULATION RESULTS

In this section, we discuss the performance of our solution and compare to that of *JFTSS* [7], *TMCP* [9] and *RBCA* [6]. The algorithms are evaluated in terms of the following metrics:

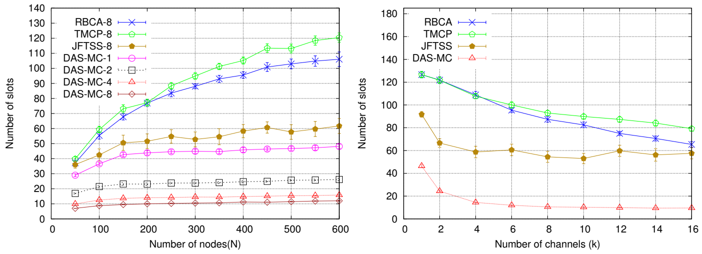
- *Time latency (last time slot):* is defined as the time required for the base station B to receive the aggregated data from all the sensor nodes;
- *Channel utilization variance σ^2:* This metric shows the capability of compared solutions in exploring the available channels. If the channels are evenly utilized over the network, the time slot reuse would be maximized and consequently the time latency will be significantly decreased. σ^2 is defined as the variance of random variable χ_i, such that χ_i represents the percentage of nodes which use channel i;

In the simulation experiments, N nodes are randomly deployed according to a uniform random distribution. The algorithms evaluation is performed by varying the number of nodes N and number of channels K. We conduct two types of experiments:

- Vary N and fix ψ to 50;
- Vary K while fixing N and ψ to 300 and 50 respectively;

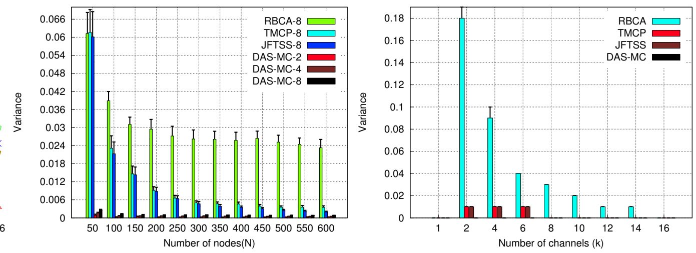
K takes one of the following values 1, 2, 4 and 8 when varying the number of nodes.

As *JFTSS*, *TMCP* and *RBCA* do not meet the data freshness requirement and in order to fairly compare *DAS-MC*



(a) (b)

Figure 2: Comparison of the data delivery latency of *DAS-MC* and the baseline approaches



(a) (b)

Figure 3: Comparing the variability in channel utilization among the compared approaches and under varying network parameters

with them, we have slightly modified their implementations so that the data freshness would be ensured. Basically, when the parent node is ready to be scheduled, it is assigned a time slot later than all its children. In our simulation results, each plotted point represents the average of 35 executions. We plot the 95% confidence interval on the graphs.

5.1 Time latency

Figure 2 shows time latency as a function of N and K. Figure 2(a) shows that when 8 channels are used and the number of nodes is 600, the time latency of *TMCP*, *RBCA* and *JFTSS* is more than 120, 105 and 62 slots, respectively, whereas *DAS-MC* does not exceed 12 slots if 8 channels are used. Figure 2(c) shows that our solution *DAS-MC* outperforms the other solution whatever the number of channels K. *DAS-MC* outperforms all baseline approaches in term of data latency due to the use of (i) intertwining the tree formation and node scheduling, and (ii) the selection criteria for choosing parent nodes. It is worth noting that *DAS-MC* achieves a lower latency even when only one channel is available, which demonstrates the effectiveness avoiding gaps and increased time slot reuse.

5.2 Channel utilization variance σ^2

Balanced use of the available channels enables better utilization of the available transceiver's capacity and decreases the time latency. Figure 3 shows the variance of channels distribution among network nodes as a function of N and K. Figures 3(a) and 3(c) show that for all values of N and K, *DAS-MC* equitably distributes the channels over the network nodes (i.e., σ^2 is not more than 0.0018) whereas σ^2 in *RBCA*, *TMCP* and *JFTSS*, in some cases, is more than 0.07, 0.05 and 0.05, respectively. Given the significantly low variance for our approach, it is not visible in Figure 3(a).

6. CONCLUSION

In time-sensitive applications minimal delay should be experienced in delivering data to the base-station. To achieve this objective, time-based medium access arbitration is often pursued and the transmission of the individual sensor nodes are scheduled in a breadth first order according to the aggregation tree. Each node gets scheduled (i.e., assigned a time slot and a channel) only after all its children. However, such an approach, which is pursued in previously published solutions, significantly constrains the optimization of the time latency. To deal with this issue, we have proposed *DAS-MC* which provides two main features: (i) it intertwines the tree construction and nodes scheduling so that an optimal time-slot and channel assignment can be achieved, and (ii) it associates parents to nodes on the aggregation tree in a way that maximizes the choices of parents for each node and maximizes time slot reuse. The simulations results have shown that *DAS-MC* scales very well with respect to number of nodes and number of channels, and outperforms the contemporary solutions in terms of time latency.

7. REFERENCES

[1] M. Bagaa, A. Derhab, N. Lasla, A. Ouadjaout, and N. Badache. Semi-structured and unstructured data aggregation scheduling in wireless sensor networks. In *Proc. of IEEE INFOCOM*, March 2012.

[2] X. Chen, X. Hu, and J. Zhu. Minimum data aggregation time problem in wireless sensor networks. volume 3794 of *Lecture Notes in Computer Science*, pages 133–142. Springer, 2005.

[3] D. Dudek, C. Haas, A. Kuntz, M. Zitterbart, D. KrÃijger, P. Rothenpieler, D. Pfisterer, and S. Fischer. A wireless sensor network for border surveillance. In *Proc. of the 7th ACM Conference on Embedded Networked Sensor Systems*, Berkeley, 2009.

[4] M. Hefeeda and M. Bagheri. Forest fire modeling and early detection using wireless sensor networks. *Ad Hoc & Sensor Wireless Networks*, 7(3-4):169–224, 2009.

[5] S. C.-H. Huang, P.-J. Wan, C. T. Vu, Y. Li, and F. F. Yao. Nearly constant approximation for data aggregation scheduling in wireless sensor networks. In *Proc. of IEEE INFOCOM*, Anchorage , AK, May 2007.

[6] Ö. D. Incel, A. Ghosh, B. Krishnamachari, and K. Chintalapudi. Fast data collection in tree-based wireless sensor networks. *IEEE Trans. Mob. Comput.*, 11(1):86–99, 2012.

[7] X. Lin and S. Rasool. A distributed joint channel-assignment, scheduling and routing algorithm for multi-channel ad-hoc wireless networks. In *Proc. of IEEE INFOCOM*, Anchorage, Alaska, May 2007.

[8] B. Malhotra, I. Nikolaidis, and M. Nascimento. Aggregation convergecast scheduling in wireless sensor networks. *Wireless Networks*, 17(2):3319–335, 2011.

[9] Y. Wu, J. A. Stankovic, T. He, and S. Lin. Realistic and efficient multi-channel communications in wireless sensor networks. In *Proc. of IEEE INFOCOM*, Phoenix, AZ, 2008.

[10] M. Younis and S. Bushra. Efficient distributed medium access arbitration for multi-channel wireless sensor networks. In *Proc. of IEEE ICC*, Glasgow, Scotland, 2007.