IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), June 2013, Darmstadt, Germany

Reducing Aggregation Bias and Time in Gossiping-based Wireless Sensor Networks

Zhiliang Chen, Alexander Kuehne, and Anja Klein
Communications Engineering Lab, Technische Universität Darmstadt, Germany
Email: {z.chen,a.kuehne,a.klein}@nt.tu-darmstadt.de

***Abstract*—Wireless sensor networks are able to perform an aggregation of the data generated by sensors. In networks where no gateway or no central sensor is specified, gossiping algorithms are used such that sensors in the whole network can aggregate messages from all other sensors. In the gossiping algorithm, the bias problem limits the quality of the aggregation results and the lack of message identification results in large aggregation time. In this paper, we reveal the possibility of eliminating or reducing the bias at the sensors by using the concept of the divisible functions that are generally applied in a sensor network and by using the memory of the sensors. Furthermore, we show how the aggregation time can be reduced by using different communication strategies for sensors communicating with their neighbors. Simulation results show the reduction of the aggregation bias at sensors as well as a higher speed of the aggregation in the network.**

I. Introduction

Wireless sensor networks (WSNs) are application-oriented networks where sensors measure data from the physical world and generate messages for aggregation [1]. A basic goal of WSNs is to aggregate messages of all sensors and perform functions on them [2], [3]. One way of doing this is to set up a sink or gateway and then build a routing tree rooted at the sink and branched out to all sensors [4]. In routing-based WSNs, sensors receive messages from other sensors, perform computations to all the messages and forward the computation output to other sensors along the route.

An alternative solution which eliminates a central sink in the network is to use random gossiping where sensors aggregate the messages based on the communications between sensors and their neighbor sensors [5]. Examples can be found in swarming and consensus applications which have been thoroughly discussed recently [5], [6]. In this paper, we consider the random gossiping algorithm with which sensors are randomly waked up to exchange messages with its neighbor sensor(s). In consensus problems, the goal of random gossiping is to asymptotically approach the average value of the measurements at each sensor [5], [7]. In [8], random gossiping with broadcasting is applied in a WSN with sparse samples at sensors to aggregate the messages of the whole network at each sensor.

In this paper, we are focusing on the idea that all sensors are capable of aggregating the messages of the entire network. We base this idea on the applications with a type of functions which have been studied in [2] and are referred to *divisible functions*. Divisible functions can be calculated distributively at the sensors in the network and they include some most common functions we are applying in a WSN, e.g. summation, averaging, max, min, histogram, etc.. Moreover, in [9], the authors argue that the summation function can be applied to calculate any function with appropriate pre-and-post processing. Therefore, the divisible functions can in a more general way calculate any function in WSNs, in a distributed way.

A problem in gossiping based communication paradigms is the bias of the aggregation at each sensor. The messages from certain sensors may be aggregated many times more than those from other sensors, as the messages exchanged between sensors in the gossiping algorithm are identity-less and the communication and aggregation are always based on local information, i.e., the information of a sensor and its surrounding neighbor sensors. What is more, in a random topology WSN, certain sensors may be located in a position where messages from other sensors are easily repeatedly aggregated. The bias problem in gossiping also results in a long aggregation (convergence) time and a large number of communications between sensors.

In this paper, we propose methods to reduce the bias of the aggregation by using messages that sensors may store in their buffers and to lessen the number of communications that are required to finish the aggregation by introducing limited redundant bits when sensors wake up and communicate with its neighbors. The effect of introducing such redundant bits is also considered.

In Section II, we give the network model as well as the notations. In Section III, we shortly discuss the divisible functions and some of their properties. In Section IV, we propose two different ways to reduce the bias of the aggregation and the aggregation time. Section V shows performance results and compares the ideas we propose to the conventional random gossiping approach. Section VI concludes this paper.

II. Network model and Notations

We consider a WSN with N randomly deployed sensors. The set of sensors is denoted by $V = \{v_1, v_2, \cdots v_N\}$. In this paper, whether there is a connection between two sensors is determined by their distance. Let d_{ij} denote the distance between sensors v_i and v_j and let d_c be a distance threshold. If $d_{ij} \leq d_c$, sensors v_i and v_j are connected, else not. $\mathcal{N}_i$ denotes the set of neighbor sensors of v_i, i.e., the set of sensors having connections to v_i.

Throughout this paper, we use the term *data* to indicate the information generated at sensors by measurements. Sensors

This work was performed within the LOEWE Priority Program Cocoon (www.cocoon.tu-darmstadt.de), supported by the LOEWE research initiative of the state of Hesse/Germany.

perform computations to the data they generated and received from other sensors and generate *messages* which indicate the bit-sequence output from computations, i.e., there may be data from several sensors in one message. The messages will be transmitted and received by sensors. We use the term *aggregation of messages* to indicate that the computations are performed to the data in the messages. The data is also referred to *parameters* of functions in the context of divisible functions.

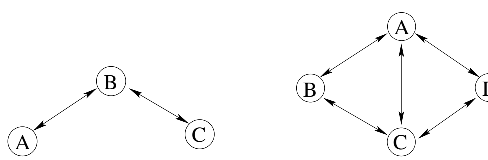
III. Divisible functions and bias of the aggregation

In this section, we discuss the divisible functions and the bias of the aggregation in gossiping. It will be shown how the bias of the aggregation can be removed based on the concept of the divisible functions.

We denote the data generated at sensor v_i by s_i. An application in WSNs corresponds to a set F of divisible functions [2]. Each divisible function $f_l \in F$ has l parameters and the functions $f_1,\ f_2,\ f_3 \cdots$ form the set F. Let $S_i, i = 1, \cdots, L$ denote disjoint non-empty sets whose elements are chosen from the parameter sets $S = \{s_1, s_2, \cdots s_K\}$, i.e., $S_i \subset S$. Let vector $\boldsymbol{s}_{S_i}$ denote the parameters given in S_i and vector $\boldsymbol{s}$ denotes all parameters in S. One property of divisible functions is that for the parameter set S and any partition $\Pi(S) = \{S_1, S_2, \cdots, S_L\}$ of it there exists a function $g^{\Pi(S)}$ such that

$$f_K(\boldsymbol{s}_S) = g^{\Pi(S)}(f_{l_1}(\boldsymbol{s}_{S_1}), f_{l_2}(\boldsymbol{s}_{S_2}), \cdots f_{l_L}(\boldsymbol{s}_{S_L})), \qquad (1)$$

where $l_i, i = 1, \cdots, L$ denotes the number of parameters in subset $S_i, i = 1, \cdots, L$. With this property, the divisible functions in WSNs can be calculated in a *divide-and-conquer fashion* [2].



(a) (b)

Fig. 1: Network examples to illustrate the bias of the aggregation

In random gossiping, sensors wake up themselves and are waked-up by neighbor sensors randomly. In examples such as the consensus problem, the identities of the data are not preserved because they cannot be distinguished after applying functions to them. An example illustrating the bias problem is shown in Fig. 1a. After sensor A and sensor B exchanging messages, both sensors have aggregated messages containing data s_A and s_B. Afterwards, sensor B may exchange messages with sensor C and both will aggregate messages containing data s_A, s_B and s_C. When sensor A and sensor B exchange messages for the second time, sensor A will receive an aggregated message containing s_A, s_B and s_C which is further aggregated with its own message which already contains $\{s_A, s_B\}$. The aggregation at sensor A will be biased since it is performed to the data multiset $\{s_A, s_B, s_A, s_B, s_C\}$. Another reason which may cause a bias of the aggregation may be loops in the network topology as shown in Fig. 1b. Even without the same pair of sensors communicating multiple times, a sensor may aggregate duplicated data due to the richly connected network topology.

In [10], it is shown that in consensus problems where only the average values is of interest in the whole network, the idea of topology control could be applied to balance the communication cost, e.g. energy consumption and the aggregation time, with bias tolerance. However, bias reduction is not considered in [10].

In this paper, we use the concept of divisible functions to reduce and even in some cases eliminate the bias. For two sensors v_C and v_R, let S_C and S_R be their sets of parameters of functions f_{l_C} and f_{l_R}, respectively, where f_{l_C} and f_{l_R} belong to the divisible function set F. We denote $\boldsymbol{s}_{S_C}$ and $\boldsymbol{s}_{S_R}$ as the parameters in S_C and S_R, respectively. If $S_C \cap S_R \neq \phi$, the aggregation

$$f_{(l_C+l_R)}(\boldsymbol{s}_{S_C}, \boldsymbol{s}_{S_R}) = g^{\Pi(\{S_C,S_R\})}(f_{l_C}(\boldsymbol{s}_{S_C}), f_{l_R}(\boldsymbol{s}_{S_R})) \qquad (2)$$

is biased. Define a set $\Psi_{CR} = \{S_1, S_2, \cdots S_\psi\}$ where ψ is the number of parameter sets in Ψ_{CR}. In order to eliminate the bias in (2), sensors v_C and v_R apply the operation $\amalg$ to the parameter set in set Ψ_{CR}, where the operation $\amalg$ applies either the unions $\cup$ or the intersections $\cap$ to the sets, such that

$$S_B = \amalg_{i=1}^{\psi} S_i = S_C \cap S_R. \qquad (3)$$

It shall be noted that, although sensors v_C and v_R both have the bias aggregation given in (2), they may have different sets Ψ_{CR}. A toy example to illustrate the operation $\amalg$ is given as follows. Assume that sensor v_C with the parameter set $S_C = \{s_1, s_2, s_3\}$ is communicating with sensor v_R whose parameter set is $S_R = \{s_3, s_4\}$. The bias exists due to $S_C \cap S_R = \{s_3\}$. If there is a set $\Psi_{CR} = \{S1, S2, S3\}$ with $S_1 = \{s_1\}$, $S_2 = \{s_2\}$ and $S_3 = \{s_1, s_2, s_3\}$, the operation $\amalg$ to get S_B is $\amalg_{i=1}^{3} S_i = (S_1 \cup S_2) \cap S_3$. For the general case, the operation of $\amalg$ getting S_B from Ψ_{CR} is given by the pseudo code in Fig. 2.

```
S_output = S_1; i = 2;
while S_output ≠ S_B do
    if S_i ∩ S_output = φ then
        S_output = S_i ∪ S;
    else
        if S_i ⊂ S_output then
            S_output = S_output ∩ S_i;
        end if
    end if
    i = i + 1;
    if i > ψ then
        i = 1;
    end if
end while
```

Fig. 2: The operation of $\amalg$

We further define $S_A = S_C \setminus S_B$. The aggregation

$$f_{l_C}(\boldsymbol{s}_{S_C}) = g^{\Pi(\{S_A,S_B\})}(f_{l_A}(\boldsymbol{s}_{S_A}), f_{l_B}(\boldsymbol{s}_{S_B})) \qquad (4)$$

is followed with a function $g^{-\Pi(\{S_C,S_B\})}$ such that

$$f_{l_A}(\boldsymbol{s}_{S_A}) = g^{-\Pi(\{S_C,S_B\})}(f_{l_C}(\boldsymbol{s}_{S_C}), f_{l_B}(\boldsymbol{s}_{S_B})). \qquad (5)$$

The unbiased aggregation of the parameters in S_C and S_R is therefore achievable by

$$f_{(l_A+l_R)}(\boldsymbol{s}_{S_A}, \boldsymbol{s}_{S_R}) = g^{\Pi(\{S_A,S_R\})}(f_{l_A}(\boldsymbol{s}_{S_A}), f_{l_R}(\boldsymbol{s}_{S_R})). \qquad (6)$$

We name the set Ψ_{CR} as the *bias-elimination set* of parameter set $S_C \cap S_R$. Instead of proving the general existence of the function $g^{-\Pi(\{S_C,S_B\})}$, in this paper, we exemplify $g^{-\Pi(\{S_C,S_B\})}$ for several functions mentioned in [2].

1) In application which calculates the *mean* of the messages, the output $f_{l_A}(\boldsymbol{s}_{S_A})$ can be calculated by

$$\begin{aligned} f_{l_A}(\boldsymbol{s}_{S_A}) &= g^{-\Pi(\{S_C,S_B\})}(f_{l_C}(\boldsymbol{s}_{S_C}), f_{l_B}(\boldsymbol{s}_{S_B})) \\ &= \frac{l_C f_{l_C}(\boldsymbol{s}_{S_C}) - l_B f_{l_B}(\boldsymbol{s}_{S_B})}{l_C - l_B} , \end{aligned} \tag{7}$$

hence the unbiased aggregation $f_{(l_A+l_R)}(\boldsymbol{s}_{S_A}, \boldsymbol{s}_{S_R})$ is

$$f_{(l_A+l_R)}(\boldsymbol{s}_{S_A}, \boldsymbol{s}_{S_R}) = \frac{l_A f_{l_A}(\boldsymbol{s}_{S_A}) + l_R f_{l_R}(\boldsymbol{s}_{S_R})}{l_A + l_R} . \tag{8}$$

2) When the *sum* function is to apply to the messages, we simply have

$$f_{l_A}(\boldsymbol{s}_{S_A}) = f_{l_C}(\boldsymbol{s}_{S_C}) - f_{l_B}(\boldsymbol{s}_{S_B}) , \tag{9}$$

and

$$f_{(l_A+l_R)}(\boldsymbol{s}_{S_A}, \boldsymbol{s}_{S_R}) = f_{l_A}(\boldsymbol{s}_{S_A}) + f_{l_R}(\boldsymbol{s}_{S_R}) . \tag{10}$$

IV. Gossiping of sensors with indicating headers

In this section, we propose methods for gossiping based WSNs for bias reduction or elimination and for reducing the aggregation time. For that purpose, we introduce an extra header which will be paired with each message and will be exchanged prior to the transmission of each application message.

For a WSN with N sensors, the *indicating header* of an aggregated message is an N-bit message field and is denoted by $\boldsymbol{I}_i$, where the subscript i indicates the relation to sensor v_i. If the current message of sensor v_i has aggregated the data generated from the measurement at sensor v_j, the j-th bit in $\boldsymbol{I}_i$, $\boldsymbol{I}_i(j)$ is marked 1, otherwise 0. The indicating header will only represent whether the corresponding data has been aggregated without showing the duplication, therefore, it corresponds to the parameter set S_i which is introduced in Section III before (1). We define an invertible function Θ which maps the parameter set S_i to the indicating header $\boldsymbol{I}_i$, i.e., $\boldsymbol{I}_i = \Theta(S_i)$ and $S_i = \Theta^{-1}(\boldsymbol{I}_i)$. Due to the existence of the biased aggregation, the parameter set S_i does not tell how many times the data from a certain sensor has been aggregated, but only which data has been aggregated.

In this paper, we assume when two sensors v_i and v_j are waked up to exchange messages, they first exchange the indicating headers of their messages and decide whether a transmission on a direction, i.e., v_j to v_i or v_i to v_j is needed. Sensor v_j will only transmit its message to v_i if $\boldsymbol{I}_j$ and $\boldsymbol{I}_i$ indicate that $S_j \not\subseteq S_i$, i.e., v_j has aggregated data that v_i has not. After sensor v_j sending its message to sensor v_i, v_i will update its indicating header as

$$\begin{aligned} \boldsymbol{I}'_i &= \Theta\left(S_i \cup S_j\right) \\ &= \Theta\left(\Theta^{-1}\left(\boldsymbol{I}_i\right) \cup \Theta^{-1}\left(\boldsymbol{I}_j\right)\right) . \end{aligned} \tag{11}$$

The same procedure is applied when sensor v_i transmits its message to sensor v_j. The gossiping algorithm stops in the network when $\boldsymbol{I}_i(j) = 1$ for $j = 1 \cdots N$ and $i = 1 \cdots N$.

In the following, we discuss two different properties of sensors leading to a reduction of aggregation bias and time. Firstly, we consider that sensors can memorize previously received messages and the concept of the divisible functions discussed in Section III which can be applied to reduce or eliminate the bias. Secondly, we consider that sensors may follow different strategies with which they communicate with their neighbors, which can be used to decrease the aggregation time.

1) Bias reduction through memorizing: First, we consider a sensor's ability to memorize the previous received messages. This considers that real sensors have buffers which can store an amount of messages together with their indicating headers. For a finite length buffer, the input-output strategy of the buffer is First-In-First-Out (FIFO).

At a certain time instant t, the newest message in the buffer of sensor v_i is the current message whose indicating header is $\boldsymbol{I}_i$. When sensor v_i receives a message from sensor v_j, it uses the indicating header $\boldsymbol{I}_i$ of its own newest message and the indicating header $\boldsymbol{I}_j$ of the received message to check the bias of the aggregation, $S_B = S_i \cap S_j = \Theta^{-1}(\boldsymbol{I}_i) \cap \Theta^{-1}(\boldsymbol{I}_j)$. If S_B is non-empty, sensor v_i uses the messages in its buffer to find the bias-elimination set Ψ_{ij} of S_B. Sensor v_i applies exhausted search method to test all combinations of the messages in its buffer. For a set of parameter sets given by a combination, sensor applies the operation given in Fig. 2. If the operation outputs the bias parameter set S_B, the given combination is then a bias-elimination set.

If sensor v_i cannot find the bias-elimination set Ψ_{ij}, sensor v_i will then ignore the bias and perform a biased aggregation. In this case, the biased aggregation will be propagated when sensor v_i communicates with other sensors.

In order to quantify the bias for measuring the performance after aggregation of messages in the whole WSN, we define $\boldsymbol{r}_i$ as an aggregation recorder at sensor v_i. $\boldsymbol{r}_i$ is a vector with integer elements of length N where the j-th entry in $\boldsymbol{r}_i$ indicates how many times parameter s_j has been aggregated in the newest message of sensor v_i. Therefore, vector $\boldsymbol{r}_i$ and the indicating-header $\boldsymbol{I}_i$ have the following relation:

$$\boldsymbol{I}_i(j) = \begin{cases} 1 & \text{if } \boldsymbol{r}_i(j) > 0 \\ 0 & \text{if } \boldsymbol{r}_i(j) = 0 . \end{cases} \tag{12}$$

We define the matrix $\boldsymbol{R}$ which is a vertical stack of $\boldsymbol{r}_i, i = 1, 2, \cdots, N$ when the gossiping algorithm stops in the network. The bias of the aggregation in the WSN is denoted by b and is defined as the normalized summation of all elements of the matrix $\boldsymbol{R}$, i.e.,

$$b = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} R(i,j) . \tag{13}$$

With this definition, when there is no bias, i.e., all entries in $\boldsymbol{R}$ are 1, the bias is $b = 1$.

2) Time reduction through message exchange strategies: Secondly, we consider a sensor's ability to use different strategies of exchanging messages with its neighbors. We apply two communication strategies which are mentioned in [5], [7] when a sensor wakes up. Note that [5], [7] focus on consensus problems to compute the average value in a WSN and do not work on gossiping for general function computations based on message exchanges with indicating headers. In the first strategy, when a sensor v_i wakes up, it exchanges messages with one neighbor sensor $v_j \in \mathcal{N}_i$. In the second strategy, the awake sensor wakes up all its neighbor sensors in $\mathcal{N}_i$ and perform time-division based messages exchanges with all of them. We name the first type of sensors the *humble sensors* and the latter type of sensors the *greedy sensors*.

The goal of both strategies is to avoid unnecessary communications in order to reduce the total number of communications in the network. When sensor v_i wakes up, it triggers all sensors in $\mathcal{N}_i$ to transmit their indicating-headers to v_i.

- If sensor v_i is a humble sensor, it chooses the sensor v_l whose indicating header results in a maximum mutual difference in the parameter sets S_i and S_l, i.e.,

$$v_l = \arg\max_{v_j \in \mathcal{N}_i} \boldsymbol{I}_i \mathrm{XOR}^b \boldsymbol{I}_j \ , \tag{14}$$

  where XOR^b performs the bit-element XOR operation and sums all elements in the resulting sequence.
- If sensor v_i is a greedy sensor, we deploy the protocol that sensor v_i is firstly a greedy listener such that it receives all messages from sensors in $\mathcal{N}_i$ in a time-division mode. Then it switches to a greedy speaker and broadcasts the aggregated messages such that all sensors in $\mathcal{N}_i$ can update their parameter set by receiving the message from v_i.

From the point of view of a practical sensor network, it is also important to consider in both strategies how long a sensor has to stay awake before the aggregation finishes in the network because a larger awake time will drain the battery of sensors faster and hence decrease their lifetime. However, this aspect is not considered in this paper and is left for future works.

V. Performance Results

In simulations, we randomly deploy $N = 20$ and $N = 30$ sensors in a two-dimensional square area, respectively. The communication range d_c of each sensor is defined such that the network remains connected. To do so, we use the concept of connectivity introduced in [10] with the Laplacian matrix of the network and its second smallest eigenvalue λ_2 to adjust d_c such that $\lambda_2 > 0$ which guarantees that no sensor or no group of senors is isolated from the rest of the sensors, respectively.

1) Buffer size of sensors vs. the bias: The probabilities of finding a bias-elimination set increases with increasing buffer size of the sensor memory.

In Fig. 3, we depict the relation between the bias of the aggregation and the buffer size. As it is shown, both greedy and humble sensors can reduce or even eliminate the bias of the aggregation by increasing the memory size of sensors. In order to zoom in the performance of the greedy and humble sensors in the figure, we do not depict the bias performance of the standing gossiping algorithm. The bias for standard random gossiping approach without memory is up to $b = 10^{10}$ on average for both $N = 20$ and $N = 30$, respectively. With the same buffer size, greedy sensors result in larger bias in comparison to humble sensors. An explanation for this is that the message which neighbor sensors in $\mathcal{N}_i$ receive from sensor

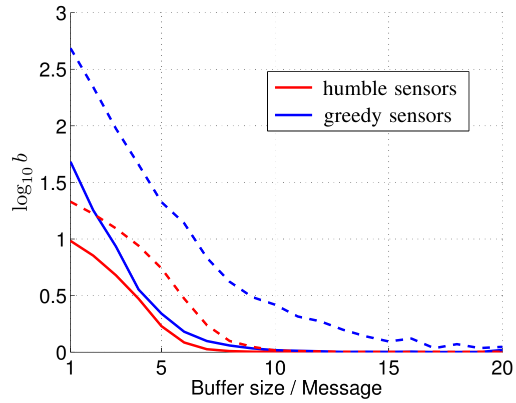


Fig. 3: Buffer size versus bias, solid lines for $N = 20$, dashed lines for $N = 30$

v_i in the greedy case contains the parameters aggregated from all sensors among $v_i \cup \mathcal{N}_i$. Therefore, more buffer is required to find the bias-elimination set. Furthermore, it is shown that with more sensors in the network which results in more neighbor sensors in $\mathcal{N}_i$, a larger buffer size is required at each sensor to find the bias-elimination set.

2) Number of Communications in different strategies: In papers regarding consensus problems using random gossiping, the convergence speed of aggregation is determined by a factor which captures the bias between the aggregation output and the true average [10]. In this paper, we assume that there are no sensors leaving or new sensors joining the network throughout the aggregation. With the indicating header, each sensor has the knowledge about how many parameters it has already aggregated. When all sensors have indicating headers whose entries are all one, the gossiping is finished. Therefore, we measure the convergence speed as the number of total number of communications of messages of all sensors.

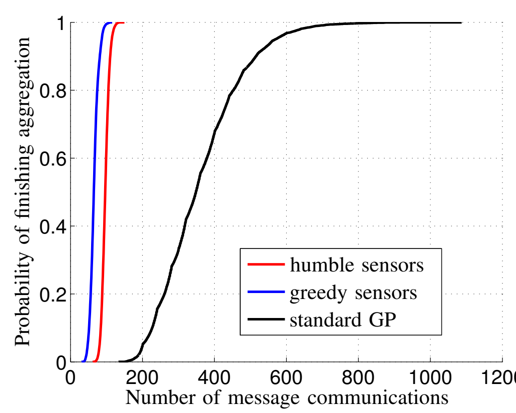


Fig. 4: Comparison of the numbers of communications required until the gossiping stops

In Fig. 4, we compare the numbers of communications required until the gossiping stops in the network, i.e., the indicating headers for messages at all sensors are all ones. This is done for the cases using indicating headers for greedy sensors (blue curves) and humble sensors (red curves). For comparison, also the performance of the Standard random GossiPing (Standard GP) discussed in [5] is shown. The abscissa in Fig. 4 is the number of message communications that are performed in the network. The ordinate gives the probability that the aggregation has been finished for all sensors in the network. Significant improvements can be witnessed by using indicating header before sensors exchanging the messages.

3) The effect of the indicating headers: In previous simulations, the additional communications for sensors exchanging the indicating headers have been neglected under the assumption that the message length in bits is much larger than N. In this part of simulations, we demonstrate the effect to the number of communications when the transmissions of the indicating-header are considered. We denote by η the ratio between the bit length of indicating-header and the length of the messages, with the assumption that all aggregations will result in the same message length in bits [11]. By adding the number of communications for exchanging the indicating headers times η to the number of communications for exchanging the messages, we can include the effect of indicating headers into our results.

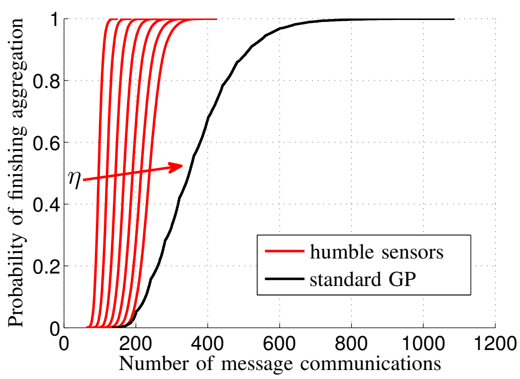


Fig. 5: Effect of indicating headers with humble sensors. From left to right, $\eta = 0\%, 5\%, 10\%, 15\%, 20\%, 25\%, 30\%$

In Fig. 5, we demonstrate the effect of indicating headers with different η when sensors are humble. Similar results can be seen in Fig. 6 when sensors are greedy. Both figures show that the gain in reducing the number of communications when considering the effect of indicating header can still be obtained even with larger η. Furthermore, the greedy sensor strategy is more efficient in terms of aggregation due to its faster spreading of messages within $v_i \cup \mathcal{N}_i$ for every v_i.

VI. Conclusion

In this paper, we considered the scenario where sensors in a wireless sensor network are aggregating messages from all other sensors using gossiping. We discussed how the concept of divisible functions can reduce the bias of the aggregation. Furthermore, we enable sensors to use messages in the memory to eliminate the bias. Two possible communication strategies have been investigated, greedy and humble. We introduced the concept of indicating header with which faster aggregation in the WSN can be achieved. Simulation results showed faster

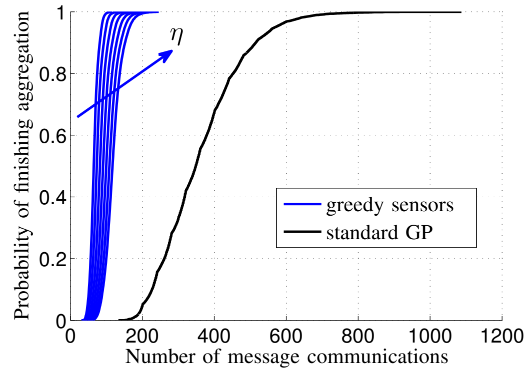


Fig. 6: Effect of indicating headers with greedy sensors. From left to right, $\eta = 0\%, 5\%, 10\%, 15\%, 20\%, 25\%, 30\%$

aggregation and significantly reduced bias in comparison to standard gossiping.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks,” *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, Aug. 2002.

[2] A. Giridhar and P. R. Kumar, “Computing and communicating functions over sensor networks,” *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 755 – 764, 2005.

[3] ——, “Toward a theory of in-network computation in wireless sensor networks,” *IEEE Communication Magazine*, vol. 44, no. 4, pp. 98–107, 2006.

[4] R. Rajagopalan and P. K. Varshney, “Data-aggregation techniques in sensor networks: A survey,” *IEEE Communications Survey and Tutorials*, vol. 8, no. 4, pp. 48 – 63, 2006.

[5] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, “Randomized gossip algorithms,” *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2508 – 2530, 2006.

[6] P. Di Lorenzo, S. Barbarossa, and A. H. Sayed, “Bio-inspired swarming for dynamic radio access based on diffusion adaptation,” in *Proc. 20th European Signal Processing Conference*, 2011.

[7] T. C. Aysal, M. E. Yildiz, A. D. Sarwate, and A. Scaglione, “Broadcast gossip algorithms for consensus,” *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2748 – 2761, 2009.

[8] J. Choi, S. Li, X. Wang, and J. Ha, “A general distributed consensus algorithm for wireless sensor networks,” in *2012 Wireless Advanced*, 2012.

[9] S. Stanczak, M. Goldenbaum, R. Cavalcante, and F. Penna, “On in-network computation via wireless multiple-access channels with applications,” in *Proc. International Symposium on Wireless Communication Systems*, 2012.

[10] S. Sardellitti, S. Barbarossa, and A. Swami, “Optimal topology control and power allocation for minimum energy consumption in consensus networks,” *IEEE Transactions on Signal Processing*, vol. 60, no. 1, pp. 383–399, 2012.

[11] Z. Chen, A. Kuehne, and A. Klein, “Delay constraints for multiple applications in wireless sensor networks,” in *Proc. IEEE International Symposium on Wireless Communication Systems*, 2012.