Improved Bias Cancellation and Header Communication for Random Gossiping-based Static 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*—Random gossiping provides a communication paradigm for wireless sensor networks so that all sensors can aggregate messages from the entire network without specifying a routing tree or a central unit. Random gossiping leads to a robust aggregation, however, it also causes biased aggregation and long aggregation time in terms of the number of communications among sensors. In our previous work, a scheme is proposed to reduce and even eliminate the aggregation bias and to reduce the number of communications by introducing indicating headers to the messages that are communicated in the network. In this paper, we extend our work assuming a static wireless sensor network. Exploiting that the topology remains constant in static networks, we introduce an improved bias cancellation method which increases the probability to achieve a bias-free aggregation and an algorithm to reduce the number of communications for indicating headers. Simulation results show the reduction of both the number of communications for indicating headers and the bias in comparison to our previous work.**

I. Introduction

Random Gossiping provides a communication paradigm in Wireless Sensor Networks (WSNs) without a central unit [1] [2] [3] [4] [5]. In this communication paradigm, sensors randomly wake up one after another, communicate messages with their neighbor sensors and switch themselves to sleep mode. In comparison with routing algorithms where only one central unit (sensor) acquires all the measurements, sensors in the network using random gossiping will aggregate measurements from all the sensors in the entire network.

Random gossiping can be applied to calculate functions in the network, especially, it achieves enormous success when applied to achieve consensus in WSNs, where only the arithmetic mean is asymptotically approached [2] [6] [7]. This is done by a sensor mixing its own measurement with the incoming measurements from its neighbor sensors with a pre-defined mixing parameter. The corresponding convergence speed can be analyzed by using the network connectivity, however, a large number of communications are required for a convergence with a controlled error.

In our previous work [8], we consider the case where random gossiping is used to compute general divisible functions in wireless sensor networks. The measurements from all sensors are the input parameters of a divisible function. The goal of the random gossiping is to communicate aggregated messages among sensors such that each measurement is included once in the aggregated message of any sensor. When a measurement is included more than once, it results in a biased aggregation. However, most commonly used divisible functions are able to reduce or even cancel the bias when the bias is identified. In [8], the idea of an Indicating Header (IH) is introduced. The IH is a communication header which marks at each sensor which measurements have been aggregated in the current message. Before the messages are communicated between sensors, the corresponding IHs are exchanged. Bias is therefore detected when the sensor compares its own current IH with the ones it receives from its neighbor sensors. The buffer at each sensor is used to store the previous messages of the sensor and their corresponding IHs. When a bias is detected, the sensor tries to construct a bias-elimination set such that the bias can be canceled [8].

In many WSNs, sensors are deployed and remain at their locations, which leads to a stable topology of the network. In the present paper, we extend our work in [8] in two aspects assuming a static WSN by exploring the fact that the topology of the WSN does not change. The first aspect is to reduce the number of communications for sensors to transmit IHs. The second aspect is to extend the bias cancellation method with a new bias reduction algorithm using so called multisets. It is worth to point out that the bias cancellation algorithm can be generally applied to both static and non-static WSNs.

The remainder of this paper is organized as follows. In Section II, we give the network model of the WSN discussed in this paper. In Section III, we shortly review our previous work in [8] and give the definition of indicating headers. In Section IV, we propose an algorithm to reduce the transmissions of the indicating headers. In Section V, the concept of multisets is shortly explained. In Section VI, the algorithm of reducing the remaining bias is introduced. Section VII shows the performance results and Section VIII concludes this paper.

II. Network Model

Let $\mathcal{V} = \{v_1, v_2, \cdots, v_N\}$ denote the set of sensors in the WSN consisting of N sensors. Each sensor is associated with a unique ID. All sensors in the network are assumed to be homogeneous such that a sensor can only be identified by its ID.

The connectivity of any two sensors $v_i \in \mathcal{V}$ and $v_j \in \mathcal{V}$, where $i \neq j$, is defined as a_{ij}, where $a_{ij} = 1$ when the two sensors are connected and 0 otherwise. In this paper, a_{ij} is solely determined by their geometrical distance d_{ij}. Let d_c be the connectivity threshold of the distance such that

$$a_{ij} = \begin{cases} 1 & d_{ij} \leq d_c \\ 0 & d_{ij} > d_c \; . \end{cases} \quad (1)$$

Sensor v_i can communicate messages with sensor v_j directly if and only if $a_{ij} = 1$. In order to guarantee the connectivity of the network, i.e., any sensor $v_i \in \mathcal{V}$ can be linked to any other sensor $v_j \in \mathcal{V}$ via one or multiple sensors in $\mathcal{V}$, the connectivity threshold d_c is lower bounded by D such that the algebraic connectivity of the graph of the network, which is defined by the second smallest eigenvalue of the Laplacian matrix of the graph, is always greater than 0 [7].

Let $\mathcal{N}_i$ denote the set of neighbor sensors of sensor v_i and N_i is the number of sensors in $\mathcal{N}_i$. Sensors in $\mathcal{N}_i$ have direct connections to v_i, i.e., $v_j \in \mathcal{N}_i$ if and only if $a_{ij} = 1$. In this paper, we consider a static WSN by assuming that during the lifetime of the network, the neighbor sensors $\mathcal{N}_i$ of every v_i remain fixed. In the remainder of this paper, we enumerate the sensors in $\mathcal{N}_i$ and let $v_k^{\mathcal{N}_i}, k = 1, 2, \cdots, N_i$ denote the k-th sensor in $\mathcal{N}_i$.

In this paper, the term *data* is used to indicate the information generated at sensors by measurement. Let s_i denote the data that sensor v_i generates and let the set $\mathcal{S} = \{s_1, s_2, \cdots, s_N\}$ denote the collection of all data in the network. There are two objectives for the WSN to achieve in this paper. The first is to compute a function whose parameters are the data of all sensors, the second is to let all sensors know the output of the function. Sensors communicate always the aggregated data which is the output of functions corresponding to the application that is running in the WSN. Throughout this paper, we consider a type of functions called *divisible functions* which can be calculated in *divide-and-conquer* manner [9]. Let $\mathcal{F}$ denote the set of divisible functions which is defined by the application running in the WSN. Each divisible function $f_l \in \mathcal{F}$ has l parameters and the functions $f_1, f_2, \cdots, f_N$ form the set $\mathcal{F}$. For any given partition $\Pi(\mathcal{S}) = \{\mathcal{S}_1, \mathcal{S}_2, \cdots, \mathcal{S}_L\}$ of the set $\mathcal{S}$ there exists a function $g^{\Pi(\mathcal{S})}$ such that

$$f_N(\boldsymbol{s}) = g^{\Pi(\mathcal{S})}\left(f_{l_1}(\boldsymbol{s}_{\mathcal{S}_1}), f_{l_2}(\boldsymbol{s}_{\mathcal{S}_2}), \cdots, f_{l_L}(\boldsymbol{s}_{\mathcal{S}_L})\right) \;, \quad (2)$$

where vector $\boldsymbol{s}_{\mathcal{S}_k}$ denotes all the data in set $\mathcal{S}_k$ and $l_k, k = 1, 2, \cdots, L$ denotes the number of data in set $\mathcal{S}_k, k = 1, 2, \cdots, L$.

Assume that at a certain time, sensor v_i has the knowledge of the output of $f_{l_i}(\boldsymbol{s}_{\mathcal{S}_i})$, where $\mathcal{S}_i \subset \mathcal{S}$. It receives from its neighbor sensors in $\mathcal{N}_i$ their computation outputs $f_{l_1^{\mathcal{N}_i}}(\boldsymbol{s}_{\mathcal{S}_1^{\mathcal{N}_i}}), f_{l_2^{\mathcal{N}_i}}(\boldsymbol{s}_{\mathcal{S}_2^{\mathcal{N}_i}}), \cdots, f_{l_{N_i}^{\mathcal{N}_i}}(\boldsymbol{s}_{\mathcal{S}_{N_i}^{\mathcal{N}_i}})$, where $\mathcal{S}_k^{\mathcal{N}_i}$ denotes the set of data that is involved in the computation at sensor $v_k^{\mathcal{N}_i}$, $\boldsymbol{s}_{\mathcal{S}_k^{\mathcal{N}_i}}$ denotes the vector of data in $\mathcal{S}_k^{\mathcal{N}_i}$ and $l_k^{\mathcal{N}_i}$ denotes the number of data in set $\mathcal{S}_k^{\mathcal{N}_i}$. All the sets, $\mathcal{S}_i$ and $\mathcal{S}_k^{\mathcal{N}_i}, v_k^{\mathcal{N}_i} \in \mathcal{N}_i$, are disjoint. Sensor v_i computes the function

$$g^{\Pi}\left(f_{l_i}(\boldsymbol{s}_{S_i}), f_{l_1^{\mathcal{N}_i}}(\boldsymbol{s}_{\mathcal{S}_1^{\mathcal{N}_i}}), f_{l_2^{\mathcal{N}_i}}(\boldsymbol{s}_{\mathcal{S}_2^{\mathcal{N}_i}}), \cdots, f_{l_{N_i}^{\mathcal{N}_i}}(\boldsymbol{s}_{\mathcal{S}_{N_i}^{\mathcal{N}_i}})\right) \quad (3)$$

and the computation output is the new function output of sensor v_i and will be transmitted to its neighbor sensors.

As an example, we assume the data set at sensor v_i is $\mathcal{S}_i = \{s_1, s_2, s_3\}$. Sensor v_i has one neighbor sensor v_j whose data set is $\mathcal{S}_j = \{s_4, s_5\}$. The function defined by the application calculates the mean value of the data, i.e., $f_3(\boldsymbol{s}_{\mathcal{S}_i}) = 1/3(s_1 + s_2 + s_3)$, $f_2(\boldsymbol{s}_{\mathcal{S}_j}) = 1/2(s_4 + s_5)$. After sensor v_i receives $f_2(\boldsymbol{s}_{\mathcal{S}_j})$ from sensor v_j, it calculates the function

$$g^{\Pi}\left(f_3(\boldsymbol{s}_{\mathcal{S}_i}), f_2(\boldsymbol{s}_{\mathcal{S}_j})\right) = \frac{1}{3+2}\left(3f_3(\boldsymbol{s}_{\mathcal{S}_i}) + 2f_2(\boldsymbol{s}_{\mathcal{S}_j})\right)$$

which yields the mean value of the new data set $\mathcal{S}_i = \{s_1, s_2, s_3, s_4, s_5\}$ at sensor v_i.

In this paper, we use the term *message* to indicate the function output (aggregated data). One *message communication* between two sensors is defined as a successful communication between the receiving and the transmitting sensors. Given the same communication range of all the sensors in the WSN, we assume that the same amount of resources for every message communication is consumed.

III. Random Gossiping with Indicating header

Random Gossiping is a de-centralized communication paradigm where the message communications are randomly initiated at a sensor. There is no routing path pre-defined in the network. In the original random gossiping algorithm used in the consensus problem, e.g., in [2] and [6], it takes a large number of message communications until the function output at each sensor converges to the consensus. In the WSN considered in this paper, the network achieves convergence when the two objectives mentioned above are achieved by using random gossiping applying so called *greedy sensors* which are introduced in our previous work [8]. In a WSN with all greedy sensors, when a sensor initiates communications with its neighbor sensors, it firstly triggers all its neighbor sensors to transmit their messages to it. Then it broadcasts its function output to all its neighbor sensors.

In [8], we discussed the bias problem which is caused by aggregating duplicated data at a sensor. In the computation (3), we assume that all the sets S_i and $S_k^{\mathcal{N}_i}, v_k^{\mathcal{N}_i} \in \mathcal{N}_i$ are disjoint, i.e., there is no data that is involved in more than one of the data sets. The aggregated data is then called an *unbiased* message. When the condition of the disjoint sets is invalid, the message is *biased*. An explanation to the occurrence of such biased message is that in random gossiping, the initiation of the message communication between a sensor and its neighbor sensors are random, therefore, after several message communications, the data of a sensor can be randomly involved in the aggregated data of more than one other sensor as long as the network is connected. Different applications have different sensitivity to the biased data aggregation. For example, in the application that defines its function as $\max()$ or $\min()$ function, biased messages cause no harm to the results required by the application. However, in applications whose function is a download function, an averaging function, a voting function, etc., biased messages strongly jeopardizes the results required by the application.

In order to lower the probability resulting in a biased message in the function output and to reduce the number of message communications when the network achieves convergence, in [8], the concept of *indicating header* is introduced. An indicating header (IH) is a fixed length bit sequence paired with each message that is generated at sensors. For a WSN with N sensors, the IH of a message has N bits. The IH of the current message at sensor v_i is denoted by $\boldsymbol{I}_i$. If the current message of v_i has aggregated the data generated at sensor $v_j, j = 1, 2, \cdots, N$, the j-th bit in $\boldsymbol{I}_i$, $\boldsymbol{I}_i(j)$ is marked 1, otherwise 0. Therefore, the IH tells only whether the corresponding data has been involved without showing its duplication and a given sensor cannot tell whether its current message is a biased message or not purely according to the corresponding IH. An invertible function Θ is defined to map the set $\mathcal{S}_i$ to $\boldsymbol{I}_i$ with $\boldsymbol{I}_i = \Theta(\mathcal{S}_i)$ and $\mathcal{S}_i = \Theta^{-1}(\boldsymbol{I}_i)$.

In [8], before sensor v_i transmits its actual message to other sensors, it firstly transmits IH $\boldsymbol{I}_i$. The message will only be transmitted if at least one of its neighbor sensors $\mathcal{N}_i$ sends feedback to indicate that $\mathcal{S}_i = \Theta^{-1}(\boldsymbol{I}_i)$ contains new data in comparison to its own data set. This communication strategy can reduce the number of message communications. In [8], it shows that when the length of the IH, i.e., N bits, is smaller than the length of the message in bits, the network consumes a smaller number of communications to achieve convergence. Meanwhile, each sensor can reduce the possibility of outputting a biased message by using the IH in combination with the buffer capability of the sensor. We call this procedure *bias cancellation*.

We assume that a sensor has buffer to store its previous messages together with the corresponding IHs. Let ψ_i denote the number of messages stored in the buffer of sensor v_i. The corresponding data sets of the stored messages are denoted by $\mathcal{S}_1^{v_i}, \mathcal{S}_2^{v_i}, \cdots, \mathcal{S}_{\psi_i}^{v_i}$, which can be determined by applying the function Θ^{-1} to the corresponding IH, and $\Psi^{v_i} = \{\mathcal{S}_1^{v_i}, \mathcal{S}_2^{v_i}, \cdots, \mathcal{S}_{\psi_i}^{v_i}\}$ denotes the set of these data sets. When sensor v_i receives a message from a neighbor sensor $v_j \in \mathcal{N}_i$, it detects that a biased function output $g^{\Pi}\left(f_{l_i}(\boldsymbol{s}_{\mathcal{S}_i}), f_{l_j}(\boldsymbol{s}_{\mathcal{S}_j})\right)$ will result if the comparison of the data set $\mathcal{S}_i = \Theta^{-1}(\boldsymbol{I}_i)$ and the data set $\mathcal{S}_j = \Theta^{-1}(\boldsymbol{I}_j)$ shows $\mathcal{S}_i \cap \mathcal{S}_j \neq \phi$.

If there is a subset $\Psi_{ij}^{v_i} \subseteq \Psi^{v_i}$, where the subscript ij indicates that it is used for cancelling the bias raised by $\mathcal{S}_i \cap \mathcal{S}_j$, and operation $\coprod$ is applied to all data sets $\mathcal{S}_l^{v_i}$ in $\Psi_{ij}^{v_i}$, where operation $\coprod$ computes either the unions or the set-theoretic difference of the sets, such that

$$\coprod_{\mathcal{S}_l^{v_i} \in \Psi_{ij}^{v_i}} \mathcal{S}_l^{v_i} = \mathcal{S}_i \cap \mathcal{S}_j, \tag{4}$$

the set $\Psi_{ij}^{v_i}$ is called the *bias-elimination set* $\mathcal{E}$ of $\mathcal{S}_i \cap \mathcal{S}_j$.

An intuitive explanation of the bias elimination is to construct an aggregated data (message) which is identical to the aggregated data (message) resulting from the data set $\mathcal{S}_i \cap \mathcal{S}_j$. The bias can then be canceled by subtracting the corresponding parts in the aggregated messages by exploiting the properties of divisible functions [8]. Let $\boldsymbol{s}_{\mathcal{S}_i \cap \mathcal{S}_j}$ denote all data in $\mathcal{S}_i \cap \mathcal{S}_j$ and l_{ij} denote the number of data in it. The operation $\coprod_{\mathcal{S}_l^{v_i} \in \Psi_{ij}^{v_i}} \mathcal{S}_l^{v_i}$ can be expressed in iterative fashion as shown in Fig. 1. The function output $f_{l_{ij}}(\boldsymbol{s}_{\mathcal{S}_i \cap \mathcal{S}_j})$ can also

$\mathcal{S}^{\mathrm{ITR}} := \phi$
for $\mathcal{S}_l^{v_i}$ in $\Psi_{ij}^{v_i}$ **do**
  $\mathcal{S}^{\mathrm{ITR}} := \mathcal{S}^{\mathrm{ITR}} \coprod \mathcal{S}_l^{v_i}$;
end for

Fig. 1. Iterative operation of $\coprod$

be calculated iteratively in such a way that when $\coprod$ applies the union operation to $\mathcal{S}^{\mathrm{ITR}}$ and $\mathcal{S}_l^{v_i}$, it applies

$$g^{\Pi}\left(f_{l^{\mathrm{ITR}}}(\boldsymbol{s}_{\mathcal{S}^{\mathrm{ITR}}}), f_{l_l^{v_i}}(\boldsymbol{s}_{\mathcal{S}_l^{v_i}})\right), \tag{5}$$

where l^{ITR} is the number of data in $\mathcal{S}^{\mathrm{ITR}}$. When $\coprod$ applies the set-theoretic difference operation to $\mathcal{S}^{\mathrm{ITR}}$ and $\mathcal{S}_l^{v_i}$, it applies

$$g^{-\Pi}\left(f_{l^{\mathrm{ITR}}}(\boldsymbol{s}_{\mathcal{S}^{\mathrm{ITR}}}), f_{l_l^{v_i}}(\boldsymbol{s}_{\mathcal{S}_l^{v_i}})\right). \tag{6}$$

The unbiased message of aggregating data in $\mathcal{S}_i$ and data in $\mathcal{S}_j$ can be achieved by

$$g^{-\Pi}\left(g^{\Pi}\left(f_{l_i}(\boldsymbol{s}_{\mathcal{S}_i}), f_{l_j}(\boldsymbol{s}_{\mathcal{S}_j})\right), f_{l_{ij}}(\boldsymbol{s}_{\mathcal{S}_i \cap \mathcal{S}_j})\right), \tag{7}$$

where $g^{-\Pi}$ is the inverse function of g^{Π}.

For example, the data set of sensor v_i is $\mathcal{S}_i = \{s_1, s_2, s_3, s_4\}$. It receives a message $f_{l_j}(\boldsymbol{s}_{\mathcal{S}_j})$ from sensor v_j, where the data set is $\mathcal{S}_j = \{s_3, s_4, s_5, s_6\}$ and $l_j = 4$. Assume that sensor v_i has stored in its buffer 4 messages whose corresponding data sets are $\mathcal{S}_1^{v_i} = \{s_1, s_2, s_3, s_4\}$, $\mathcal{S}_2^{v_i} = \{s_1, s_2, s_4\}$, $\mathcal{S}_3^{v_i} = \{s_2, s_4\}$, $\mathcal{S}_4^{v_i} = \{s_2\}$. The bias is detected as $\mathcal{S}_i \cap \mathcal{S}_j = \{s_3, s_4\} \neq \phi$. The bias-elimination set is $\Psi_{ij}^{v_i} = \{\mathcal{S}_1^{v_i}, \mathcal{S}_2^{v_i}, \mathcal{S}_3^{v_i}, \mathcal{S}_4^{v_i}\}$ because with the elements of $\Psi_{ij}^{v_i}$ one can construct $((\mathcal{S}_1^{v_1} - \mathcal{S}_2^{v_1}) \cup \mathcal{S}_3^{v_1}) - \mathcal{S}_4^{v_1} = \mathcal{S}_i \cap \mathcal{S}_j$. If the function defined by the application is to calculate the mean value, the biased aggregation is performed to the aggregated data $f_4(\boldsymbol{s}_{\mathcal{S}_i}) = 1/4(s_1 + s_2 + s_3 + s_4)$ and $f_4(\boldsymbol{s}_{\mathcal{S}_j}) = 1/4(s_3 + s_4 + s_5 + s_6)$ yielding

$$\begin{aligned} & g^{\Pi}\left(f_{l_i}(\boldsymbol{s}_{\mathcal{S}_i}), f_{l_j}(\boldsymbol{s}_{\mathcal{S}_j})\right) = \frac{1}{8}(4f_4(\boldsymbol{s}_{\mathcal{S}_i}) + 4f_4(\boldsymbol{s}_{\mathcal{S}_j})) \\ = \; & \frac{1}{8}\left(s_1 + s_2 + 2s_3 + 2s_4 + s_5 + s_6\right) . \end{aligned}$$

By applying the algorithm in Fig. 1 to the data set in $\Psi_{ij}^{v_i}$, one constructs $f_{l_{ij}}(\boldsymbol{s}_{\mathcal{S}_i \cap \mathcal{S}_j}) = 1/2(s_3 + s_4)$. The unbiased message is then calculated by

$$\begin{aligned} & g^{-\Pi}\left(g^{\Pi}\left(f_{l_i}(\boldsymbol{s}_{\mathcal{S}_i}), f_{l_j}(\boldsymbol{s}_{\mathcal{S}_j})\right), f_{l_{ij}}(\boldsymbol{s}_{\mathcal{S}_i \cap \mathcal{S}_j})\right) \\ = \; & \frac{1}{6}\left(8g^{\Pi}\left(f_{l_i}(\boldsymbol{s}_{\mathcal{S}_i}), f_{l_j}(\boldsymbol{s}_{\mathcal{S}_j})\right) - 2f_{l_{ij}}(\boldsymbol{s}_{\mathcal{S}_i \cap \mathcal{S}_j})\right) \\ = \; & \frac{1}{6}\left(s_1 + s_2 + s_3 + s_4 + s_5 + s_6\right) . \end{aligned}$$

IV. Reduction of IH communications in Static WSNs

In [8], the number of message communications is reduced by transmitting IHs prior to the messages. However, it leads to a large number of communications for transmitting IHs. In this paper, we exploit the advantage of a static network to significantly reduce the number of IH transmissions.

In the considered random gossiping, there are two types of communications in which a sensor $v_i \in \mathcal{V}$ can participate. In the first type, sensor v_i initiates the message communications with all its neighbor sensors. In this type, v_i firstly triggers all sensors in $\mathcal{N}_i$ to send their messages to v_i, v_i generates the function output with all data involved in the received messages and the message of its own and then broadcasts the aggregated data to all sensors in $\mathcal{N}_i$. In the second type, sensor v_i belongs to the neighbor sensors of another sensor which initiates the message communications. In this type, the message transmission of sensor v_i is not a broadcast but only a point-to-point transmission.

In a static network, the neighbor sensors of every sensor remain constant in the network. Therefore, it is reasonable to assume that the neighbor sensors are able to know the information of the message at sensor v_i if sensor v_i has not participated in any communications of the second type after the previous first type communication initiated by sensor v_i. We propose an algorithm in Fig. 2 to utilize this knowledge for reducing the number of IH transmissions.

It can be seen from the algorithm shown in Fig. 2 that the message which sensor v_i broadcasts to sensors in $\mathcal{N}_i$ to initiate the message communications contains one bit to indicate whether there is new data updated in the data set $\mathcal{S}_i$ since last type one communication. According to this 1-bit information, sensors in $\mathcal{N}_i$ acquire $\boldsymbol{I}_i$ by either receiving it from v_i if the bit is 1 or by recovering it from the previously received message if the bit is 0.

v_i initiates the message communication;
v_i broadcasts to sensors in $\mathcal{N}_i$ a message indicating the communication initiation;
v_i informs sensors in $\mathcal{N}_i$ whether it has new data updated in the date set of $\mathcal{S}_i$ of v_i since last type one communication;
if there is new data updated in the data set of $\mathcal{S}_i$ of v_i since last type one communication **then**
    v_i broadcasts $\boldsymbol{I}_i$ to $\mathcal{N}_i$;
else
    sensors in $\mathcal{N}_i$ recover $\boldsymbol{I}_i$ which is received from the previous type one communication initiated by v_i;
end if
for v_j in $\mathcal{N}_i$ **do**
    if message of v_j contains data that is new to v_i **then**
        v_j feed backs its $\boldsymbol{I}_j$ to v_i;
    else
        if message of v_i contains data that is new to v_j **then**
            v_j informs v_i that it requires message communication from v_i;
        else
            v_j transmits nothing;
        end if
    end if
end for
v_i processes all feedbacks;
v_i informs sensors in $\mathcal{N}_i$ which sensors need to send their messages;
All informed neighbor sensors transmit their messages;
v_i computes the new aggregated data;
if new aggregated data at v_i contains new data for sensors in $\mathcal{N}_i$ **then**
    v_i broadcasts the new $\boldsymbol{I}_i$;
    v_i broadcasts the new message;
end if

Fig. 2. Algorithm to reduce IH communications

V. Multiset Representation

Before introducing the improved bias-cancellation algorithm, we introduce the concept of *multiset* which provides a tool to define the bias and the operations related to the bias-cancellation more conveniently. A multiset includes the multiplicity of a data in the data set and can this can be used to handle the multiplicity of a data in the function computations. A multiset is formally defined as a 2-tuple $(\mathcal{A}, m_\mathcal{A})$ where $\mathcal{A}$ is some set and $m_\mathcal{A}$ is a function which maps the elements in $\mathcal{A}$ to positive natural numbers. For each $\alpha \in \mathcal{A}$, the multiplicity, i.e., the number of occurrences, of α is represented by the number $m_\mathcal{A}(\alpha)$. For two multisets $(\mathcal{A}, m_\mathcal{A})$ and $(\mathcal{B}, m_\mathcal{B})$, the summation operation $(\mathcal{A}, m_\mathcal{A}) + (\mathcal{B}, m_\mathcal{B})$ is defined by

$$
\begin{aligned}
& (\mathcal{C}, m_\mathcal{C}) = (\mathcal{A}, m_\mathcal{A}) + (\mathcal{B}, m_\mathcal{B}) \; , \\
\text{where} \quad & \mathcal{C} = \mathcal{A} \cup \mathcal{B} \quad \text{and} \\
& m_\mathcal{C}(s) = \begin{cases} m_\mathcal{A}(s) + m_\mathcal{B}(s) & \forall s \in \mathcal{A} \cap \mathcal{B} \\ m_\mathcal{A}(s) & \forall s \in \mathcal{A} - (\mathcal{A} \cap \mathcal{B}) \\ m_\mathcal{B}(s) & \forall s \in \mathcal{B} - (\mathcal{A} \cap \mathcal{B}) \end{cases}
\end{aligned}
$$

where s is a set element.

The subtraction operation of multiset $(\mathcal{A}, m_\mathcal{A}) - (\mathcal{B}, m_\mathcal{B})$ can only be applied when the following two conditions are fulfilled,

- $\mathcal{B} \subseteq \mathcal{A}$ and
- $\forall \alpha \in \mathcal{B}, \; m_\mathcal{A}(\alpha) - m_\mathcal{B}(\alpha) \geq 0$.

Under these two conditions, the subtract operation $(\mathcal{C}, m_\mathcal{C}) = (\mathcal{A}, m_\mathcal{A}) - (\mathcal{B}, m_\mathcal{B})$ can be defined such that

$$
\begin{aligned}
& (\mathcal{C}, m_\mathcal{C}) = (\mathcal{A}, m_\mathcal{A}) - (\mathcal{B}, m_\mathcal{B}) \qquad (8) \\
\text{where} \quad & m_\mathcal{C}(s) = \begin{cases} m_\mathcal{A}(s) & s \in \mathcal{A} - \mathcal{B} \\ m_\mathcal{A}(s) - m_\mathcal{B}(s) & s \in \mathcal{B} \end{cases} \\
\text{and} \quad & s \in \mathcal{C} \quad \text{if} \quad m_\mathcal{C}(s) > 0
\end{aligned}
$$

If in a multiset $(\mathcal{A}, m_\mathcal{A})$, the occurrence of all elements in $\mathcal{A}$ are 1, the multiset can be written as $(\mathcal{A}, \mathbf{1})$, indicating that all data in $\mathcal{A}$ has the multiplicity being 1.

VI. Improved bias cancellation in Static WSNs

In [8], the bias cancellation is applied at sensor v_i with each received message of its neighbor sensors individually. In this paper, we propose an algorithm for bias cancellation at v_i which jointly processes *all* incoming messages from $\mathcal{N}_i$. When sensor v_i receives all feedbacks from $\mathcal{N}_i$, i.e., the IHs, it can calculate the data set of the new aggregated data by comparing the data of each data set outputted from the function Θ^{-1}. We assume that after receiving or recovering $\boldsymbol{I}_i$, sensors in a subset of $\mathcal{N}_i$, denoted by $\mathcal{N}_i^{\mathrm{s}}$, feed back their IHs to v_i. In the following, we denote N_i^{s} as the number of sensors in $\mathcal{N}_i^{\mathrm{s}}$ and let $v_k^{\mathcal{N}_i^{\mathrm{s}}}, k = 1, 2, \cdots, N_i^{\mathrm{s}}$ denote the sensors in $\mathcal{N}_i^{\mathrm{s}}$ whose data sets are $\mathcal{S}_k^{\mathcal{N}_i^{\mathrm{s}}}, k = 1, 2, \cdots, N_i^{\mathrm{s}}$, respectively.

A biased message may result if v_i computes the function output with all the messages from sensors in $\mathcal{N}_i^{\mathrm{s}}$. Let $\mathcal{S}_i^{\mathrm{R}}$ denote the data set if all data in $\mathcal{S}_k^{\mathcal{N}_i^{\mathrm{s}}}, k = 1, 2, \cdots, N_i^{\mathrm{s}}$ are aggregated, i.e.,

$$\mathcal{S}_i^{\mathrm{R}} = \cup_{k=1}^{N_i^{\mathrm{s}}} \mathcal{S}_k^{\mathcal{N}_i^{\mathrm{s}}} \cup \mathcal{S}_i \ , \tag{9}$$

where $\mathcal{S}_i^{\mathrm{R}}$ is termed *the reference data set of aggregation* at sensor v_i. When represented by a multiset, it is denoted by $(\mathcal{S}_i^{\mathrm{R}}, \mathbf{1})$.

Defining the set $\Psi_i^{\mathcal{N}_i^{\mathrm{s}}} = \left\{\mathcal{S}_k^{\mathcal{N}_i^{\mathrm{s}}}, k = 1, 2, \cdots, N_i^{\mathrm{s}}\right\}$ and the subset $\Psi \subseteq \Psi_i^{\mathcal{N}_i^{\mathrm{s}}}$, we represent every data set $\mathcal{S}_k \in \Psi_i^{\mathcal{N}_i^{\mathrm{s}}}$ with multiset $(\mathcal{S}_k, m_{\mathcal{S}_k})$ and calculate the multiset sum $\sum_{\mathcal{S}_k \in \Psi} (\mathcal{S}_k, m_{\mathcal{S}_k}) + (\mathcal{S}_i, m_{\mathcal{S}_i})$, where the output is denoted by $(\mathcal{S}_\Psi, m_{\mathcal{S}_\Psi})$. For a subset $\Omega \subseteq \Psi^{v_i}$, we apply the operation $\coprod$ to all sets in Ω yielding $(\mathcal{S}_\Omega, m_{\mathcal{S}_\Omega})$ and the multiset subtraction $(\mathcal{S}_\Psi, m_{\mathcal{S}_\Psi}) - (\mathcal{S}_\Omega, m_{\mathcal{S}_\Omega})$ yielding $(S_{\Psi-\Omega}, m_{S_{\Psi-\Omega}})$. If $\sum_{\alpha \in S_{\Psi-\Omega}} (m(\alpha) - 1)$ is the minimum among all possible subset of Ψ^{v_i}, the Ω is termed *bias-reduction set* of $(\mathcal{S}_\Psi, m_{\mathcal{S}_\Psi})$ and the summation output is termed the *remaining bias* of $(\mathcal{S}_\Psi, m_{\mathcal{S}_\Psi})$. Since several data sets are involved to calculate the function output at sensor v_i, it is possible that $m(\alpha) - 1 > 1, \alpha \in S_{\Psi-\Omega}$. It is then necessary to further find a bias reduction set of $(S_{\Psi-\Omega}, m_{S_{\Psi-\Omega}})$ to further reduce the remaining bias. This procedure is performed iteratively until no further bias reduction can be done.

Let b denote the *bias-cancellation function* to calculate the remaining bias after all iterations with the input set of data set $\Psi \subseteq \Psi_i^{\mathcal{N}_i^{\mathrm{s}}}$. The problem of finding the optimal Ψ, which is denoted by $\Psi^{\min_i}$, can be formulated as

$$\begin{aligned} \Psi_i^{\min} \quad &= \quad \arg \min_{\Psi \subseteq \Psi_i^{\mathcal{N}_i^{\mathrm{s}}}} b(\Psi) \\ \text{s.t.} \quad & \quad \cup_{\mathcal{S}_k \in \Psi} \mathcal{S}_k = \mathcal{S}_i^{\mathrm{R}} \ . \end{aligned} \tag{10}$$

In order to solve the problem in (10) efficiently, we firstly partition all data set in $\Psi_i^{\mathcal{N}_i^{\mathrm{s}}}$ and group them according to their relationship, i.e., a data set is a subset or a superset of another. Let $\mathcal{P}$ denote the set of groups after partitioning and let p denote the number of groups in $\mathcal{P}$. Let $\mathcal{P}_j, j = 1, 2, \cdots, p$ denote the j-th group in $\mathcal{P}$. In group $\mathcal{P}_j$, let variable $n_{\mathcal{P}_j}$ denote the current number of data sets and let $\mathcal{P}_j(l), l = 1, 2, \cdots, n_{\mathcal{P}_j}$ denote the l-th data set in $\mathcal{P}_j$. All the groups in $\mathcal{P}$ are ordered, i.e., the first data set of a group is a superset of all other data sets in the group. Let $\mathcal{P}^1$ denote the set of the first data sets of all groups in $\mathcal{P}$. The algorithm in Fig. 3 shows the procedure to find $\mathcal{P}^1$.

```
P^1 is initialized to be P^1 = {S_1^{N_i^s}}
for S_k^{N_i^s} in Ψ_i^{N_i^s} do
    join_P^1 := 0
    for S_k in P^1 do
        if S_k^{N_i^s} ⊆ S_k then
            join_P^1 := 0;
            Stop current for-loop and start with the next S_k^{N_i^s};
        else
            if S_k^{N_i^s} ⊃ S_k then
                join_P^1 := 1;
                P^1 = P^1 − S_k;
            else
                join_P^1 := 1;
            end if
        end if
    end for
    if join_P^1 = 1 then
        P^1 = P^1 ∪ {S_k^{N_i^s}};
    end if
end for
```

Fig. 3. Algorithm to find first data set of all groups in $\mathcal{P}$

In $\mathcal{P}$, the number p of groups is set to be the number of data sets in $\mathcal{P}^1$ in the algorithm of Fig. 3. The union of all data sets in $\mathcal{P}^1$ is identical to $\mathcal{S}_i^{\mathrm{R}}$, the reference data set of aggregation at sensor v_i. The algorithm in Fig. 4 shows how to do the partitioning and grouping knowing $\mathcal{P}^1$.

For example, the data sets in $\Psi_i^{\mathcal{N}_i^{\mathrm{s}}}$ are $\{\mathcal{S}_1^{\mathcal{N}_i^{\mathrm{s}}}, \cdots, \mathcal{S}_6^{\mathcal{N}_i^{\mathrm{s}}}\}$, where $\mathcal{S}_1^{\mathcal{N}_i^{\mathrm{s}}} = \{s_1, s_2, s_3\}$, $\mathcal{S}_2^{\mathcal{N}_i^{\mathrm{s}}} = \{s_2, s_3, s_4\}$, $\mathcal{S}_3^{\mathcal{N}_i^{\mathrm{s}}} = \{s_4, s_5\}$, $\mathcal{S}_4^{\mathcal{N}_i^{\mathrm{s}}} = \{s_1, s_2\}$, $\mathcal{S}_5^{\mathcal{N}_i^{\mathrm{s}}} = \{s_2\}$ and $\mathcal{S}_6^{\mathcal{N}_i^{\mathrm{s}}} = \{s_3, s_4\}$, respectively. Set $\mathcal{P}^1$ is $\{\mathcal{S}_1^{\mathcal{N}_i^{\mathrm{s}}}, \mathcal{S}_2^{\mathcal{N}_i^{\mathrm{s}}}, \mathcal{S}_3^{\mathcal{N}_i^{\mathrm{s}}}\}$ according to the algorithm in Fig. 3 and the result of the grouping by using algorithm in Fig. 4 is $\mathcal{P}_1 = \{\mathcal{S}_1^{\mathcal{N}_i^{\mathrm{s}}}, \mathcal{S}_4^{\mathcal{N}_i^{\mathrm{s}}}, \mathcal{S}_5^{\mathcal{N}_i^{\mathrm{s}}}\}$, $\mathcal{P}_2 = \{\mathcal{S}_2^{\mathcal{N}_i^{\mathrm{s}}}, \mathcal{S}_5^{\mathcal{N}_i^{\mathrm{s}}}, \mathcal{S}_6^{\mathcal{N}_i^{\mathrm{s}}}\}$ and $\mathcal{P}_3 = \{\mathcal{S}_3^{\mathcal{N}_i^{\mathrm{s}}}\}$.

Secondly, we will choose from each group in $\mathcal{P}$ one data set and test whether their union is equivalent to $\mathcal{S}_i^{\mathrm{R}}$. If the equivalence holds, then the current choice is treated as the candidate for calculating the function output and perform the bias cancellation. Let $\mathcal{C}$ denote the collection of all these choices of candidates, let $n_{\mathcal{C}}$ denote the number of choices in $\mathcal{C}$ where each choice has p data sets chosen from each group of $\mathcal{P}$. Let $\mathcal{C}_m, m = 1, 2, \cdots, n_{\mathcal{C}}$ denote m-th choice in $\mathcal{C}$ and $\mathcal{C}_m(l), l = 1, 2, \cdots, p$ denote the l-th data set in $\mathcal{C}_m$. With the given example above, we have $\mathcal{C} = \{\mathcal{C}_1, \cdots, \mathcal{C}_4\}$, where $\mathcal{C}_1 = \{\mathcal{S}_1^{\mathcal{N}_i^{\mathrm{s}}}, \mathcal{S}_2^{\mathcal{N}_i^{\mathrm{s}}}, \mathcal{S}_3^{\mathcal{N}_i^{\mathrm{s}}}\}$, $\mathcal{C}_2 = \{\mathcal{S}_1^{\mathcal{N}_i^{\mathrm{s}}}, \mathcal{S}_6^{\mathcal{N}_i^{\mathrm{s}}}, \mathcal{S}_3^{\mathcal{N}_i^{\mathrm{s}}}\}$, $\mathcal{C}_3 = \{\mathcal{S}_4^{\mathcal{N}_i^{\mathrm{s}}}, \mathcal{S}_2^{\mathcal{N}_i^{\mathrm{s}}}, \mathcal{S}_3^{\mathcal{N}_i^{\mathrm{s}}}\}$ and $\mathcal{C}_4 = \{\mathcal{S}_4^{\mathcal{N}_i^{\mathrm{s}}}, \mathcal{S}_6^{\mathcal{N}_i^{\mathrm{s}}}, \mathcal{S}_3^{\mathcal{N}_i^{\mathrm{s}}}\}$, respectively.

Finally, we test the bias cancellation performance of each choice in $\mathcal{C}$. The bias-cancellation function b is performed for

1: **for** $\mathcal{S}_k^{\mathcal{N}_i^s}$ in $\Psi_i^{\mathcal{N}_i^s} - \mathcal{P}^1$ **do**
2: **for** $\mathcal{P}_j$ in $\mathcal{P}$ **do**
3: join_group := 0;
4: join_group_END := 1;
5: **for** $l = 1$ **to** $n_{\mathcal{P}_j}$ **do**
6: **if** $\mathcal{S}_k^{\mathcal{N}_i^s} = \mathcal{P}_j(l)$ **then**
7: join_group := 0;
8: Stop the current **for**-loop and go to line 21
9: **else**
10: **if** $\mathcal{S}_k^{\mathcal{N}_i^s} \subset \mathcal{P}_j(l)$ **then**
11: join_group := 1;
12: **else**
13: **if** $\mathcal{S}_k^{\mathcal{N}_i^s} \supset \mathcal{P}_j(l)$ **then**
14: join_group := 1;
15: join_group_end := 0;
16: Stop the current **for**-loop and go to line 21
17: **end if**
18: **end if**
19: **end if**
20: **end for**
21: **if** join_group $= 1$ **then**
22: **if** join_group_END $= 0$ **then**
23: $\mathcal{S}_k^{\mathcal{N}_i^s}$ joins the group $\mathcal{P}_j$ at the last position;
24: **else**
25: $\mathcal{S}_k^{\mathcal{N}_i^s}$ joins the group $\mathcal{P}_j$ at the l-th position;
26: **end if**
27: **end if**
28: **end for**
29: **end for**

Fig. 4. Algorithm for partitioning and grouping

each $\mathcal{C}_m \in \mathcal{C}$. The problem in (10) is now transformed to

$$\Psi_i^{\min} = \arg\min_{\mathcal{C}_m \in \mathcal{C}} b(\mathcal{C}_m), \tag{11}$$

which can be solved by exhaustive search. In comparison to the search space of the problem in (10) which is $2^{N_i^s} - 1$, the search space in (11) is $n_{\mathcal{C}}$.

After finding the optimal $\Psi_i^{\min}$, sensor v_i informs the corresponding neighbor sensors to transmit their messages. In comparison to what we proposed in [8], it is possible that only a subset of sensor v_i's neighbor sensors need to transmit their messages.

It is worth to point it out that this bias cancellation algorithm can also be applied to a non-static WSN.

VII. Simulation Results

In the simulations, we set the number of sensors in the network to $N = 50$. Fig. 5 shows an example of one topology realization with the communication range indicated by the circle. In the realization, we randomly deploy 50 sensors in a 1000-by-1000 squared area and guarantee the connectivity of the network by choosing the communication range as shown in the figure. Fig. 6 shows the number of neighbor sensors of each sensor in the realization shown in Fig. 5.

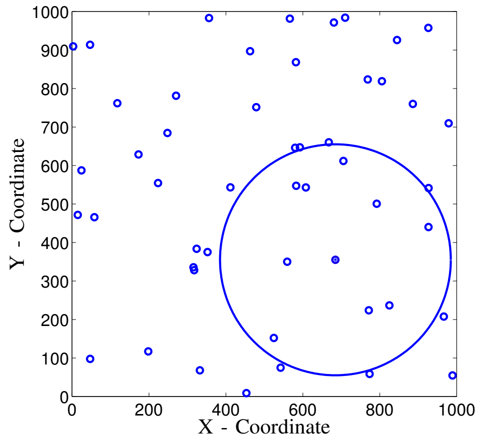

Fig. 5. A topology realization of WSN

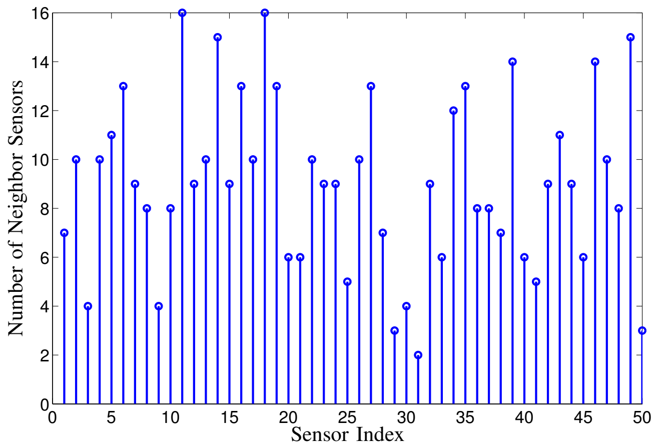

Fig. 6. Number of neighbor sensors for WSN realization in Fig. 5

In Fig. 7, for the topology realization shown in Fig. 5 and Fig. 6, the average number of communications for transmitting IHs (red lines) at each sensor with our algorithm is shown and compared with full IH communications (black lines) in [8]. Note that the 1-bit overhead indicating new aggregated data when sensor initiates message communications with its neighbor sensors is neglected. As it is shown in the figure, a huge reduction in IH transmissions is achieved by exploiting the advantage of a static network and avoiding unnecessary IH transmissions.

For the topology realization shown in Fig. 5 and Fig. 6, Fig. 8 shows the difference in remaining bias when the network achieves convergence, denoted by Δ_{bias}, by comparing the bias reduction resulting from the bias cancellation algorithm in [8] to the bias resulting from the proposed algorithm in this work. The results are topology sensitive and dependent on the sensor deployment.

In Fig. 9 and Fig. 10, we show the performance with four WSN realizations, denoted by WSN Realization-1 to WSN Realization-4, respectively. In each realization, we randomly deploy 50 sensors in a 1000-by-1000 squared area

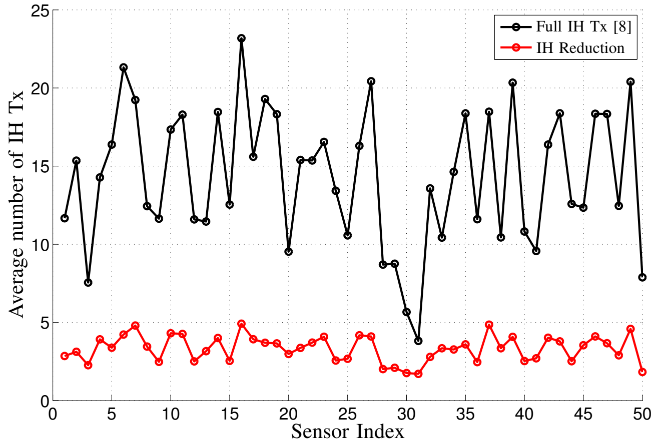

Fig. 7. Number of IH Communications

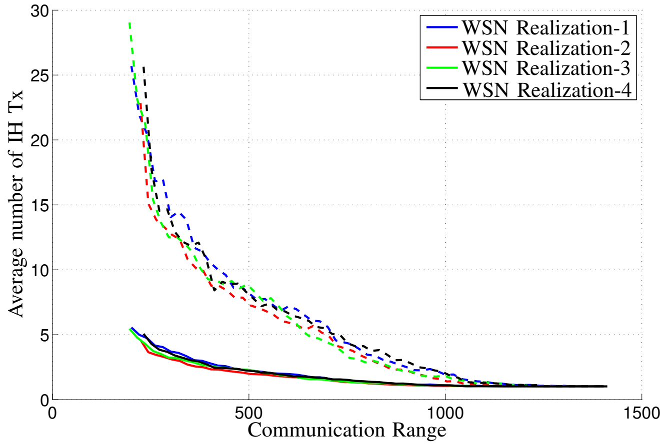

Fig. 9. Number of IH Communications vs. Communication Range

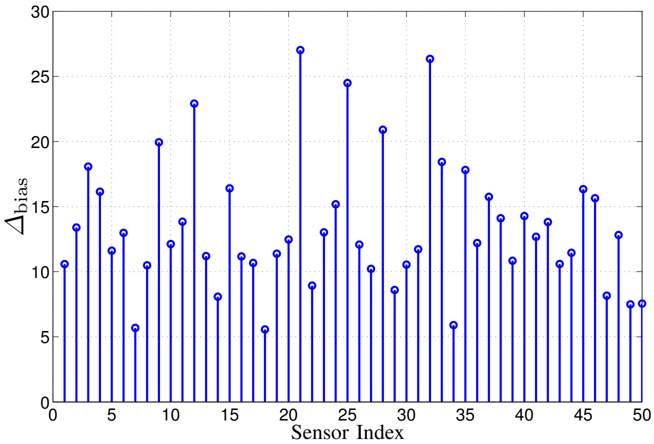

Fig. 8. Δ_{bias} comparing new algorithm to [8]

and increase the communication range of each sensor. For all four realizations, the minimum communication range with the guaranteed connectivity is around 150. When we increase the communication range, the number of neighbor sensors of each sensor will also increase. The WSN realization-1 with blue lines in both figures corresponds to the WSN from the results shown in Fig. 7 and Fig. 8.

In Fig. 9, the dashed lines show the average number of IH transmissions per sensor applying the full IH transmission scheme of [8] and the solid lines depict the number of IH transmissions of the proposed algorithm. When the communication range is increased, the gain by using the proposed algorithm in comparison to the algorithm in [8] is reduced since larger communication range leads to faster convergence of the network and the number of IH transmissions reduces for both scheme shown in the Figure. The performance of the two algorithms merges when the communication range is so large that the neighbor sensors of each sensor include almost all sensors in the network.

Fig. 10 shows the additional remaining bias that can be canceled per each sensor compared to [8]. The proposed algorithm outperforms the algorithm in [8] for communication ranges below 1000. For larger communication ranges, where the neighbor sensors of each sensor includes almost all sensors in the network, both algorithms perform equally. It is because with high probability, a sensor only need to participate once in a type one or type two communication until the network converges resulting in no biased messages.

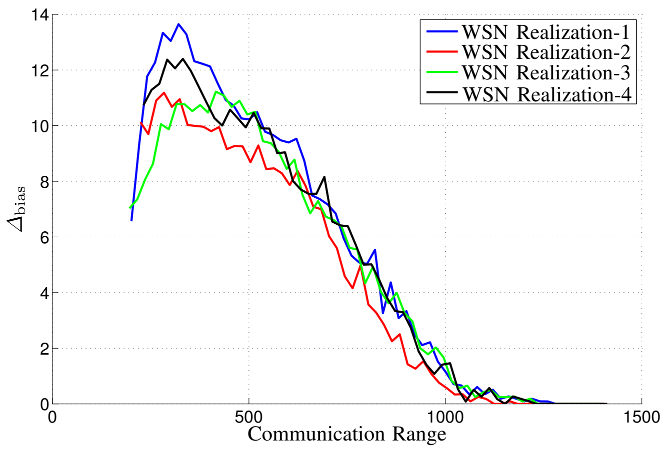

Fig. 10. Δ_{bias} vs. Communication Range

VIII. Conclusion

This paper discusses how to reduce the number of transmissions of indicating headers and how to further reduce the remaining bias in a static WSN using random gossiping. We propose algorithms exploiting the advantage of a static network where the neighbor sensors of a sensor remain constant in the network lifetime. Sensors can remember the IH information of their neighbor sensors, therefore, when the neighbor sensors have no new data aggregated, the IH does not need to be transmitted. Furthermore, a new bias cancellation algorithm, which can be applied to both static and non-static WSNs, is proposed such that a sensor can simultaneously perform the bias cancellation jointly processing several incoming messages. Simulations show both the reduction of IH transmissions and remaining bias in comparison to the results in our previous work.

IX. Acknowledgement

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.

References

[1] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," in *Proc. 44th Annual IEEE Symposium on Foundations of Computer Science*, 2003.

[2] 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.

[3] D. Mosk-Aoyama and D. Shah, "Fast distributed algorithms for computing separable functions," *IEEE Transactions on Information Theory*, vol. 54, no. 7, pp. 2997 – 3007, 2008.

[4] S. Sundaram and C. Hadjicostis, "Distributed function calculation and consensus using linear iterative strategies," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 4, pp. 650 – 660, 2008.

[5] M. Goldenbaum, H. Boche, and S. Stanczak, "Nomographic gossiping for f-consensus," in *Proc. 10th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, 2012.

[6] 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.

[7] 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.

[8] Z. Chen, A. Kuehne, and A. Klein, "Reducing aggregation bias and time in gossiping-based wireless sensor networks," in *Proc. 14th IEEE International Workshop on Signal Processing Advances in Wireless Communciations*, 2013.

[9] 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.