Multi-hop Coordination 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*—Gossiping-based wireless sensor networks provide a communication paradigm with which all sensors in the network can aggregate messages from the entire network without specifying a routing tree and a sink sensor. Random gossiping provides a robust aggregation, however, it also leads to biased aggregation and long aggregation time in terms of the number of communications between sensors. In a previous work, we proposed a scheme to reduce and even eliminate the bias of the aggregation with a smaller number of communications by introducing an indicating header to each message that is communicated in the network. In this paper, we extend our work by a multi-hop coordination of sensors such that when a sensor wakes up in a random gossiping-based sensor network, it can coordinate the message exchange with sensors which are more than one hop away. In order to measure the stability of the network topology, we introduce the failure rate reflecting how often a senor fails to perform message exchange with other sensors. We provide simulation results to show the reduction of the number of communications that are required for every sensor to aggregate all messages of the entire network.**

I. Introduction

Wireless sensor networks (WSNs) are data-centric networks where messages are exchanged and aggregated between sensors [1]. In order to aggregate the messages of all sensors, two typical techniques are applied. In the first one, a gateway is defined for the sensor network and a routing tree is built with the gateway being the root and all sensors being either leaves or branches of the tree [2]. In this configuration, the communication of the messages has only one direction, i.e., from sensors to the root. The sensors in the routing tree will receive messages from their children sensors, perform computations and forward to their father sensors. The communication stops in the network when the gateway received messages from all its children sensors.

The second technique is random gossiping. It specifies a communication paradigm in a WSN that sensors are randomly waked up and exchange messages with their neighbor sensors, i.e., the communication of messages is not constrained in one direction. With application examples in [3] and [4], random gossiping is used for all sensors to achieve consensus of their measurements. In [5], random gossiping is applied to dynamic radio access in networks. In this technique, all sensors are aggregating messages from the sensors of the entire network, i.e., the aggregated messages can actually be retrieved from any sensor in the network. Random gossiping provides a more robust communication method in WSNs against link failure and topology changes than routing because no gateway is specified and no communication path shall be maintained from each sensor to the gateway [3].

In our work, we combine the robust random gossiping algorithm introduced in [3] and [4] and the routing based message communications in [2] such that every sensor in the network will aggregate messages from the entire network. We extend the aggregation in WSNs from calculating consensus to divisible functions which include some most common functions we are interested in, e.g., summation, averaging, max, min, histogram, etc. [6]. In [7], we proposed two methods which are jointly used to reduce the aggregation bias and the aggregation time in gossiping-based WSNs. The property of divisible functions is explored for cancelling the bias of the aggregation by using the messages stored in the buffer of sensors. We introduced an indicating header to each message communicated in the network indicating whether the data of a certain sensor has been aggregated in the current message. Two strategies how a sensor can exchange messages with its neighbor sensors have been compared as well.

The goal of this paper is to further decrease the number of communications that are required for all sensors to aggregate all messages in the network in random gossiping-based WSNs by extending the schemes we proposed in [7]. This shall be achieved by enabling the coordination of message exchange with sensors which are more than one hop away. Multi-hop in random gossiping was previously discussed in [8] and [9] in consensus problems. In [8], the averaging is performed along a path which is randomly built towards another random target sensor in the network when a sensor randomly wakes up. In [9], a gossiping algorithm for consensus is preformed in a grid sensor network. A sensor can gossip with a sensor that is within several hops and determined by the gradient of the measurements at sensors. In [8] and [9], only a consensus problem is discussed, whereas our work extends the aggregation function in WSNs to all divisible functions. Furthermore, the possibility of bias reduction is not considered in [8] and [9].

In this paper, we continue our previous work in [7]. The multi-hop random gossiping based on the indicating headers of the messages in the network for calculating divisible functions is discussed. Furthermore, we consider the possibility that a sensor fails to communicate with other sensors and analyze the impact on the aggregation time.

The remainder of this paper is organized as follows. In Section II, we give the system model and the notations. In Section III, we briefly review the content of our previous work. In Section IV, we introduce the scheme of the multi-

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.

hop coordination. Section V shows the performance results and Section VI concludes this paper.

II. System Model and Notations

Throughout this paper, N denotes the number of sensors which are randomly deployed. The set of sensors is denoted by $V = \{v_1, v_2, \cdots, v_N\}$. The existence of the connection between sensors v_i and v_j is solely determined by their distance which is denoted by d_{ij}. Let d_c be a threshold such that for any two sensors $v_i \in V$ and $v_j \in V$ the connection between v_i and v_j exists when $d_{ij} \le d_c$. In order to guarantee the connectivity of the network, d_c is chosen in such a way that the Laplacian matrix of the network has a positive second smallest eigenvalue [10]. $\mathcal{N}_i$ denotes the set of neighbor sensors of sensor v_i, i.e., the set of sensors having connections to v_i.

In this paper, we use the term *data* to indicate the information generated at sensors by measurement. Sensors perform computations to the data that they generated themselves and that they received from other sensors. The term *message* indicates 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 term data is also referred to as *parameters* of functions in the context of divisible functions.

III. Bias of gossiping-based aggregation and indicating headers

In this section, we review the schemes that we introduced in [7] to reduce aggregation bias and time in gossiping-based WSNs as far as needed to understand the rest of the paper. We base our scheme of reducing the bias of aggregation on divisible functions. Let s_i denote the data generated at sensor v_i. An application in WSNs corresponds to a set F of divisible functions [6]. 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 $S_i, i = 1, 2, \cdots, L$ denote disjoint non-empty sets whose elements are chosen from the parameter set $S = \{s_1, s_2, \cdots, s_K\}$, i.e., $S_i \subset S$, which corresponds to the set of measured data from all sensors in the network. Let vector $\boldsymbol{s}_{S_i}$ denote the parameters given in set S_i and vector $\boldsymbol{s}$ denote all parameters in S. The divisible function gives a type of function in WSNs which can be calculated in a *divide-and-conquer* fashion [6]. This means that for the parameter set S and any partition $\Pi(S) = \{S_1, S_2, \cdots, S_L\}$ of it, there is 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 $S_i, i = 1, \cdots, L$.

In random gossiping, sensors wake up themselves and are waked up by their neighbor sensors randomly. Due to the lack of the identity of the data in the messages, one cannot tell which data has been included in the computation. In consequence, the output message may contain a message from a sensor several times or does not contain the message from a sensor at all. We call such aggregation as biased aggregation. For two sensors $v_C \in V$ and $v_R \in V$, their corresponding parameter sets are S_C and S_R, respectively. Functions $f_{l_C} \in F$ and $f_{l_R} \in F$ are their corresponding divisible functions, respectively. The respective parameters in S_C and S_R are $\boldsymbol{s}_{S_R}$ are $\boldsymbol{s}_{S_R}$. If $S_C \cap S_R \neq \phi$, the aggregation

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

is biased, i.e., there are parameters which have been aggregated unequally in the output.

In order to reduce the bias in the results of (2), in [7] we define the *bias-elimination set* $\Psi_{CR} = \{S_1, S_2, \cdots, S_\psi\}$ of parameter set $S_C \cap S_R$, where ψ is the number of parameter sets in Ψ_{CR}. We use the operation $\amalg$ which applies either the union $\cup$ or the intersection $\cap$ to the parameter set in the set Ψ_{CR}. The bias-elimination set fulfills

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

We define the parameter set $S_A = S_C - S_B$. 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 by a function $g^{\Pi^{-1}(\{S_C, S_B\})}$ such that

$$f_{l_A}(\boldsymbol{s}_{S_A}) = g^{\Pi^{-1}(\{S_C, S_B\})}(f_{l_C}(\boldsymbol{s}_{S_C}), f_{l_B}(\boldsymbol{s}_{S_B})). \qquad (5)$$

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

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

In [7], we also introduce the concept of an *indicating header* which is a fixed length bit sequence paired with each message that is generated and computed at sensors. For a WSN with N sensors, the indicating header of an aggregated message is an N-bit message field. The indicating header of the message of sensor v_i is denoted by $\boldsymbol{I}_i$. If the current message of sensor 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 indicating header tells only whether the corresponding data has been aggregated without showing its duplication. We use the invertible function Θ to map the parameter set S_i to the indicating header $\boldsymbol{I}_i$ with $\boldsymbol{I}_i = \Theta(S_i)$ and $S_i = \Theta^{-1}(\boldsymbol{I}_i)$.

When two sensors v_i and v_j wake up to exchange messages, they firstly exchange their indicating headers of their messages and decide whether a transmission of the message on a direction, i.e., v_j to v_i or v_i to v_j is needed [7]. The concept of indicating headers can be used to reduce the aggregation bias in combination with the property of divisible functions shown from (2) to (6). The realization is made through memorizing by considering the fact that sensors have buffers to store previous received messages. When sensor v_i receives a message from its neighbor sensor v_j, it uses 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. If sensor v_i cannot find the bias-elimination set Ψ_{ij}, sensor v_i will then neglect the bias and perform a biased aggregation. In this case, the bias aggregation will be propagated when sensor v_i performs further communications with other sensors.

IV. Multi-hop coordination

In this section, we extend our work in [7] by considering a multi-hop coordination in the message exchange process.

Our scheme of multi-hop coordination in gossiping-based WSNs is that when a sensor wakes up it can exchange messages with sensors which are several hops away. Multi-hop coordination can extend the range of sensors with which the awake sensor exchanges their messages. The motivation of such multi-hop coordination is to increase the aggregation efficiency. From the point of view of one sensor in the network, it is most efficient to set this sensor as the root and build a routing tree to connect all sensors in the network when it is about to aggregate all messages in the network. For a sensor network with N sensors, when a tree is built with sensor v_i being the root, it requires N communications of messages until sensor v_i aggregates all messages. Because the wake-up is random and the communicating sensor pairs are not scheduled, the random gossiping cannot guarantee such small number of communications for sensor v_i, even with the indicating headers we proposed in [7] who can reduce the number of unnecessary communications. However, as a communication paradigm with less topology and stability requirements than building a routing tree, random gossiping benefits from its robustness and its flexibility.

The multi-hop coordination combines the capability of a connected network for building a tree rooted at any sensor with the flexibility of a gossiping based communication paradigm. This natural extension of gossiping-based WSN is based on the fact that a sensor in a sensor network always plays two roles in terms of communication: a receiver and a transmitter.

For a sensor $v_i \in V$ in a connected WSN, the *depth* $\delta_j(i)$ of another sensor $v_j \in V$ with respect to sensor v_i is defined as the minimum number of hops with which v_i can send its message to v_j and vice versa. In the following, we define the coordination depth c_i of sensor v_i as the maximum depth that sensor v_i can coordinate message exchanges with other sensors. From this, it follows that sensor v_i can exchange message with all sensors v_j whose depth $\delta_j(i)$ is smaller than c_i when it wakes up for communications. With $\mathcal{N}_i^l$ denoting the set of sensors whose depth with respect to v_i is l, all sensors in $\cup_{l=1}^{c_i}\mathcal{N}_i^l$ are the possible sensors who can exchange messages with v_i.

Due to the failure of communications, the failure of the wake-ups, etc., the query information may not be received or responded by a sensor. We model the rate of failure with a parameter which is referred to as *failure rate* r_i of sensor v_i which indicates the probability that sensor v_i fails to receive query information or to respond to another sensor in the network.

In our previous work [7], we propose two message exchange strategies when a sensor wakes up. In the case that a sensor exchanges message with only one sensor within its neighbor sensors, we call the sensor *a humble sensor*. In the case that a sensor exchanges its messages with all the sensors within its neighborhood, we call the sensor *a greedy sensor*. When the multi-hop coordination is enabled in the network, we use the term humble sensor to indicate that the sensor v_i will coordinate a single path with length c_i. For the greedy sensor of v_i, a tree, whose root is sensor v_i, is constructed among sensors in $\cup_{l=1}^{c_i}\mathcal{N}_i^l$.

In the case when the humble sensor strategy is applied in the network, let $\mathcal{P}_i$ denote the path that is initiated by sensor v_i with the maximum possible depth being c_i and let $\mathcal{P}_i(0) = v_i$ and $\mathcal{P}_i(l)$ indicate the root sensor and the sensor on the l-th hop of this path, respectively. The awake sensor v_i broadcasts a query message to all its neighbor sensors in $\mathcal{N}_i^1$. In this query message, the information is contained that sensor v_i asks all sensors in $\mathcal{N}_i^1$ for their indicating headers and the information telling sensors in $\mathcal{N}_i^1$ that a path is going to be constructed. Sensors in $\mathcal{N}_i^1$ receive this query information and respond to it with failure rate $r_j, v_j \in \mathcal{N}_i^1$. We denote the set of sensors in $\mathcal{N}_i^1$ who successfully receive and respond to this requirement from v_i by $P(\mathcal{N}_i^1)$. Sensors $v_j \in P(\mathcal{N}_i^1)$ will send back their indicating headers $\boldsymbol{I}_j$. Sensor v_i will choose the sensor which sends back its indicating header and which results in the greatest bi-directional message differences to be its next hop. The chosen sensor broadcasts the query information to $\mathcal{N}_i^2$ and chooses its next hop. Such process continues until either the maximum coordination depth c_i is reached or no more sensors respond to join the path when c_i has not been reached. In general, the criterion to choose the sensor for the l-th hop is given by

$$\mathcal{P}_i(l) = \arg \max_{v_j \in P(\mathcal{N}_i^l) \cap \mathcal{N}_{\mathcal{P}_i(l-1)}} \boldsymbol{I}_{\mathcal{P}_i(l-1)} \mathrm{XOR}^b \boldsymbol{I}_j \ , \tag{7}$$

where the operation XOR^b performs the XOR-operation to the bit-sequence in $\boldsymbol{I}_i$ and $\boldsymbol{I}_j$ and gives the number of positive bits in the output. The algorithm of constructing path $\mathcal{P}_i$ is given in Figure 1.

```
1: P_i(0) = v_i;
2: l = 1;
3: while l ≤ c_i do
4:     P_i(l − 1) broadcasts query messages to sensors in N_i^l
5:     Determine the set P(N_i^l)
6:     Determine P_i(l) with (7)
7:     l = l + 1
8: end while
```

Fig. 1. Algorithm of constructing a path initiated by sensor v_i

The actually achieved path depth is denoted by c_i^α, where $c_i^\alpha \le c_i$. We denote v_i as the header of path $\mathcal{P}_i$ and $\mathcal{P}(c_i^\alpha)$ as the tail sensor, respectively. Once the path $\mathcal{P}_i$ is constructed, the transmission of the messages starts from the tail sensor. $\mathcal{P}(c_i^\alpha)$ transmits its message to $\mathcal{P}(c_i^\alpha - 1)$, sensor $\mathcal{P}(c_i^\alpha - 1)$ aggregates it with its own message and the combined message is transmitted to $\mathcal{P}(c_i^\alpha - 2)$. This procedure is done until sensor v_i has the aggregated message whose indicating header is

$$\boldsymbol{I}_i = \Theta\left(\cup_{l=0}^{c_i} \Theta^{-1}\left(\boldsymbol{I}_{\mathcal{P}_i(l)}\right)\right) \ . \tag{8}$$

Afterwards, sensor v_i starts to transmit its message along the path $\mathcal{P}_i$ towards the tail sensor until every sensor $v_j \in \mathcal{P}_i$ updates the aggregated message with indicating header $\boldsymbol{I}_j = \boldsymbol{I}_i$.

In the case that sensors in the network are greedy sensors, each sensor $v_i \in V$ will attempt to coordinate a tree whose root is v_i and the maximum depth is c_i. Let $\mathcal{T}_i$ be the tree rooted at sensor v_i. $\mathcal{T}_i(l)$ denotes the set of sensors whose depth with respect to sensor v_i is l and $\mathcal{T}_i(0) = v_i$. The father sensor of sensor v_j in the tree $\mathcal{T}_i$ is denoted by $\mathcal{T}_i^f(v_j)$. When sensor v_i wakes up, it broadcasts a query message with its own indicating header $\boldsymbol{I}_i$. Sensors in $\mathcal{N}_i^1$ receive the query message and respond to it with failure rate r_j, where $v_j \in \mathcal{N}_i^1$. The set of sensors who successfully receive and respond to the query is denoted by $P(\mathcal{N}_i^1)$ and hence $\mathcal{T}_i(1) = P(\mathcal{N}_i^1)$. Each sensor in $\mathcal{T}_i(1)$ continues this tree construction by forwarding the query message with its indicating header. If a sensor v_j whose depth with respect to v_i is l, i.e., $v_j \in P(\mathcal{N}_i^l), l = 2, \cdots, c_i$, receives query messages and indicating headers from several sensors, it decides itself which sensor shall be its father sensor $\mathcal{T}_i^f(v_j)$. Let $\mathcal{N}_j^{\mathcal{T}_i} = \mathcal{N}_j \cap P(\mathcal{N}_i^{l-1})$ denote the set of sensors from which sensor v_j receives indicating messages. The criterion of sensor v_j to choose its father sensor is given as

$$\mathcal{T}_i^f(v_j) = \arg \max_{v_k \in \mathcal{N}_j^{\mathcal{T}_i}} \boldsymbol{I}_k \mathrm{XOR}^b \boldsymbol{I}_j \ . \tag{9}$$

The algorithm of constructing the tree $\mathcal{T}_i$ is given in Figure 2

```
T_i(0) = v_i;
l = 1;
while l ≤ c_i do
    for v_l ∈ T_i(l − 1) do
        v_l broadcasts query message to sensors in N_i^{l+1} ∩ N_l
        Determine the set P(N_i^{l+1} ∩ N_l)
    end for
    for v_m ∈ ∪_{v_l ∈ T_i(l−1)} P(N_i^{l+1} ∩ N_l) do
        Determine N_m^{T_i}
        Determine T_i^f(v_j) with (9)
    end for
    l = l + 1
end while
```

Fig. 2. Algorithm of constructing a tree initiated by sensor v_i

The actually achieved tree depth is denoted by c_i^β, where $c_i^\beta \leq c_i$. The communication in $\mathcal{T}_i$ stars with sensors in $\mathcal{T}_i(c_i^\beta)$ transmitting their messages to their father sensors in $\mathcal{T}_i(c_i^\beta - 1)$. After a sensor in $\mathcal{T}_i(c_i^\beta - 1)$ receives messages from all its children, it forwards the aggregated messages to its father sensors. This procedure finishes until sensor v_i receives messages from all its children. Sensor v_i will have an aggregated message whose indicating header is

$$\boldsymbol{I}_i = \Theta\left(\cup_{v_j \in \boldsymbol{T}_i} \Theta^{-1}\left(\boldsymbol{I}_j\right)\right) \ . \tag{10}$$

Afterwards, sensor v_i broadcasts its newly aggregated messages which contains messages from all sensors in $\mathcal{T}_i$ to $\mathcal{T}_i(1)$ and sensors in $\mathcal{T}_i(1)$ forward this messages to their children sensors. This procedure stops when all sensors in $\mathcal{T}_i$ have received the aggregated messages from v_i. Every sensor $v_j \in \mathcal{T}_i$ will now have an updated message with indicating header $\boldsymbol{I}_j = \boldsymbol{I}_i$. Note that for both humble and greedy cases we assume that the sensors only suffer from failures in the phase of constructing paths or trees, respectively.

V. Performance Results

In this section, we demonstrate the performance of the scheme we proposed in Section IV. In the simulations, $N = 30$ sensors are randomly deployed in a two-dimensional squared area.

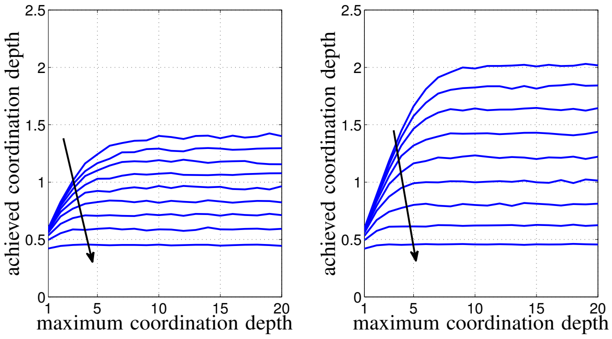

Fig. 3. Maximum coordination depth versus achieved coordination depth with failure rate. Left: humble sensors, right: greedy sensors. Along the direction of the arrow, $r_i = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9$

In Fig. 3, we depict the relation between the maximum coordination depth of all sensors in the network, i.e., $c_i, v_i \in V$ and the actually achieved coordination depth in the network under different failure rates of sensors. As shown in the figure, with both humble and greedy sensor strategies, it is unlikely that the maximum coordination depth can be achieved when the failure rate increases. In comparison, the greedy sensor strategy results in a larger achieved coordination depth due to the fact that all neighbor sensors who decide not to reject the requirement will join the tree.

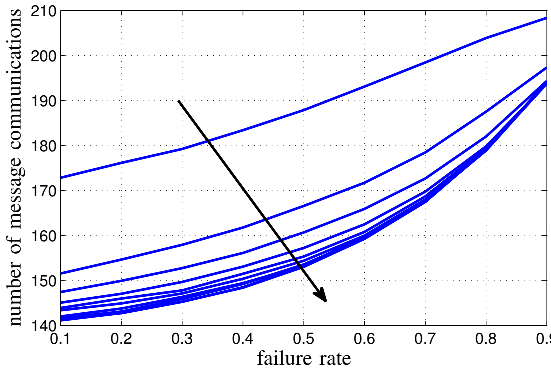

Fig. 4. Average number of message communications required in the network until the aggregation is finished for humble sensors. Along the arrow, the maximum coordination depth $c_i = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$.

In Fig. 4, we demonstrate how many message communications in the network have been performed until the aggregation is finished when all sensors are humble sensors, i.e., a path is constructed when a sensor wakes up. As shown in the figure, when the failure rate increases, more message communications are needed due to the failure of the communication. Meanwhile, increasing the maximum coordination depth decreases the number of required message communications. A significant reduction of the number of message communications can be witnessed by just increasing the coordination depth from $c_i = 1$ which corresponds to the scheme in [7] to $c_i = 2$. For coordination depths $c_i > 3$, the additional reduction by further increasing the maximum coordination depth is small. When sensors in the network are greedy sensors, i.e., a tree

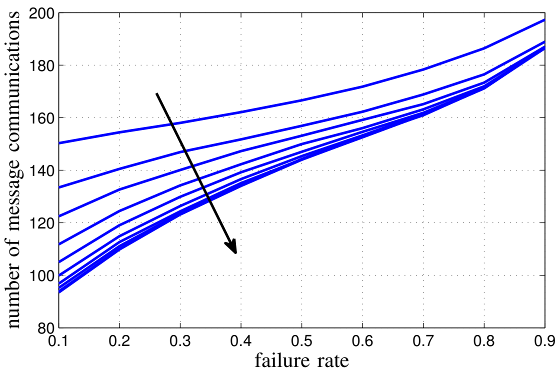

Fig. 5. Average number of message communications required in the network until the aggregation is finished for greedy sensors. Along the arrow, the maximum coordination depth $c_i = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$.

is constructed when a sensor wakes up, the performance of the number of message communications is shown in Fig. 5. In comparison to humble sensors, less message communications are needed with the greedy sensor strategy.

In order to take into consideration the communications that have to be spent to exchange indicating headers, we assume that the indicating header requires 10% of the message length. We define *equivalent communications* as the sum of the number of message communications and 0.1 times the number of communications spent for indicating headers. The performance of the humble sensor case is shown in Fig. 6 and of the greedy sensor case is shown in Fig. 7, respectively. As shown in Fig. 6, with the humble sensor strategy, a larger coordination depth still results in a lower number of communications. However, with the greedy sensor strategy, such benefit by increasing coordination depth can only be achieved with small failure rate $r_i < 0.3$. As seen in Fig. 7, with a larger failure rate, the number of equivalent communications for a larger coordination depth is even worse compared to the case with smaller coordination depths.

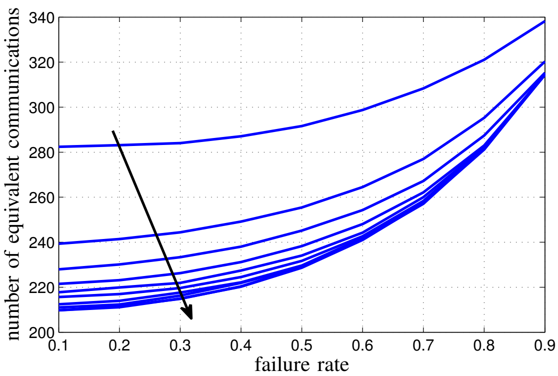

Fig. 6. Average number of equivalent communications required in the network until the aggregation is finished for humble sensors. Along the arrow, the maximum coordination depth $c_i = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$.

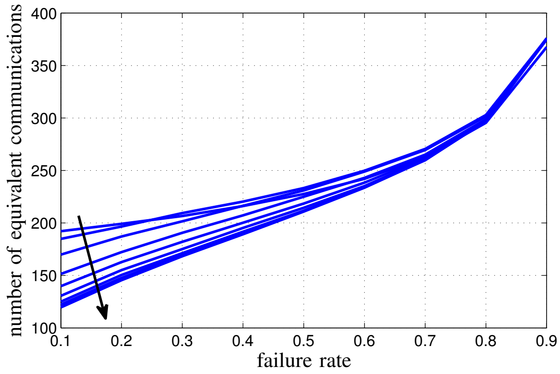

Fig. 7. Average number of equivalent communications required in the network until the aggregation is finished for greedy sensors. Along the arrow, the maximum coordination depth $c_i = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$.

VI. Conclusion

In this paper, we extend our previous work in gossiping-based wireless sensor networks. We introduce a multi-hop coordination to sensors when sensors randomly wake up to exchange messages with their neighbor sensors. To take into account the failure of communications between sensors, a failure rate of sensors is also proposed. Simulation results show the reduction of the number of communications needed for performing aggregations in the whole network with the proposed schemes.

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] 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.

[3] 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.

[4] 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.

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

[6] 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.

[7] 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.

[8] F. Benezit, A. G. Dimakis, and P. Thiran, "Order-optimal consensus through randomized path averaging," *IEEE Transactions on Information Theory*, vol. 56, no. 10, pp. 5150 – 5167, 2010.

[9] D. Ustebay, B. Oreshkin, M. Coates, and M. Rabbat, "Multi-hop greedy gossip with eavesdropping," in *Proc. International Conference on Information Fusion*, 2009.

[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.