Zhiliang Chen, Alexander Kuehne, and Anja Klein, ”Randomized Gossip Protocol in Wireless Sensor Networks with Partial Sensor Involvement,” in *Proc. the eleventh International Symposium on Wireless Communication Systems*, August 2014.

Randomized Gossip Protocol in Wireless Sensor Networks with Partial Sensor Involvement

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—**In this paper, we consider a wireless sensor network where only a part of the sensors in the network generate measurement data and have interest in the aggregation output. However, in order to maintain connectivity of the sensors involved in the application, namely application-member sensors, other so called non-application-member sensors, have to participate in the communications to assist in the message exchanges in the network. On the one hand, the application-member sensors have the objective of a fast convergence, i.e., only a small number of communications should be performed until all measurement data are aggregated at application-member sensors. On the other hand, only few non-application-member sensors should participate in the communications and only a small number of communications should be carried out by those who participate in. We propose a refined randomized gossip protocol based on our previous work to address the two mentioned problems. The results show that with the proposed approach the number of involved non-application-member sensors as well as the number of communications performed by the involved non-application-member sensors are both significantly reduced compared to the approach in our previous work.**

I. INTRODUCTION

Sensors in Wireless Sensor Networks (WSN) are capable of measuring vast types of data, which could be of interest to different applications. There are situations where not all sensors in the network are needed to participate in the application, e.g., a sensor network runs multiple applications [1] [2] [3]. In this case, one application may only require the support of part of the sensors in the network. There are various dependencies to choose sensors for an application, e.g., based on the location information of the sensors telling whether they are near to phenomenon hotspots, on the result of slicing algorithm or on the remaining energy of a sensor, etc.

This paper considers one application running in a WSN in which only a subset of sensors are participating, namely application-member sensors. They generate measurement data and aim at aggregating measurement data from all sensors who are also involved in the same application. It is possible that these application-member sensors cannot directly communicate with each other, hence sensors who are not involved in the applications, namely, non-application-member sensors, are obliged to assist in the communications to enable successful data aggregations among the application-member sensors. Since all application-member sensors have interest in the aggregation output, randomized gossip is chosen as a reasonable candidate communication protocol.

Randomized gossip is known as a communication paradigm which enables sensors in WSNs to perform aggregations without specifying a central control node [4] [5] [6]. Without the central control unit, aggregations in the network are based on local communications which are randomly initialized by sensors. However, for most randomized gossip algorithms, aggregating all data at all sensors requires a large number of communications due to the randomness and redundant message transmissions. In our previous works [7], [8] and [9], we introduced indicating headers (IH) to randomized gossip in wireless sensor networks to reduce the number of communications.

The contribution of this paper is to introduce a refined randomized gossip protocol applied to WSNs where only a part of the sensors in the network are involved in one application and the output of a divisible function should be known to the involved sensors. The refined randomized gossip categorizes six different scenarios together with different protocols to reduce the number of communications performed by non-application-member sensors and to decrease the number of involved non-application-member sensors.

The remainder of this paper is organized as follows. In Section II, we show the network model. In Section III, the idea of indicating headers for randomized gossip is shortly reviewed. In Section IV, we introduce the refined randomized gossip protocol which works on WSNs with only a subset of sensors being application-member sensors. Section V discusses the simulation results and the conclusion is given in Section VI.

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. There are two types of sensors in the WSN, the application-member sensors which are denoted by $\mathcal{V}_\mathrm{A} \subset \mathcal{V}$ and the non-application-member sensors which are denoted by $\mathcal{V}_\mathrm{K} = \mathcal{V} \setminus \mathcal{V}_\mathrm{A}$. We assume that the application-member sensors generate measurements and perform computations to them, communicate and store the computation output and have interests in the aggregation output. The non-application-member sensors are not involved in the application, therefore

978-1-4799-5863-4/14/$31.00 ©2014 IEEE

they should save its energy by only assisting the communications between the application-member sensors. We assume all sensors in $\mathcal{V}$ to have the same communication range d. The connectivity of the network is guaranteed by a lower bound of d which yields the second smallest eigenvalue of the Laplacian matrix of the network to be greater than 0 [10]. Let $\mathcal{N}_i$ denote the set of neighbor sensors which fall in the communication range of sensor v_i and have direct connections to v_i. Furthermore, $\mathcal{N}_i^{\mathrm{A}}$ denotes the application-member neighbors of sensor v_i and $\mathcal{N}_i^{\mathrm{K}}$ the non-application-member neighbors. A static WSN is assumed in this paper such that during the lifetime of the network, the neighbor sensors $\mathcal{N}_i$ of every v_i remain constant.

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 $v_i \in \mathcal{V}_{\mathrm{A}}$ generates and let the set $\mathcal{S} = \{s_i | v_i \in \mathcal{V}_{\mathrm{A}}\}$ denote the collection of all data from the application-member sensors 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 application-member sensors. The second is to let all application-member sensors know the output of the function. Sensors communicate 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 [11].

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 transmitting and receiving sensor.

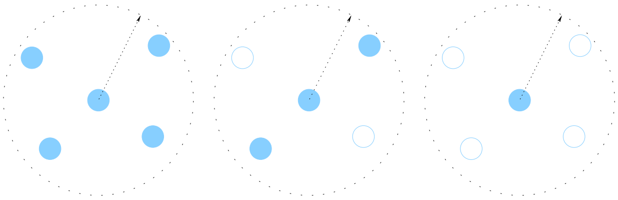
III. INDICATING HEADER

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. An invertible function Θ is defined to map the data set $\mathcal{S}_i$ of v_i to $\boldsymbol{I}_i$ with $\boldsymbol{I}_i = \Theta(\mathcal{S}_i)$ and $\mathcal{S}_i = \Theta^{-1}(\boldsymbol{I}_i)$.

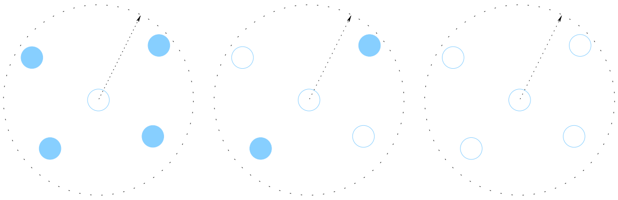
In [7], before sensor v_i transmits its actual message to other sensors, it firstly transmits IH $\boldsymbol{I}_i$. The message will only be transmitted if at least one of its neighbor sensors $\mathcal{N}_i$ sends feedback to indicate that $\mathcal{S}_i = \Theta^{-1}(\boldsymbol{I}_i)$ contains new data in comparison to its own data set. In [7], it is shown that introducing IH to the randomized gossip protocol can reduce the number of message communications when the network achieve convergence in comparison to the randomized gossip protocol where no IH is utilized.

IV. REFINED RANDOMIZED GOSSIP PROTOCOL

Let $\mathcal{V}_{\mathrm{B}} \subseteq \mathcal{V}_{\mathrm{K}}$ denote the non-application-member sensors which assist in the message communications. Let N_{A}, N_{K} and N_{B} denote the number of sensors in $\mathcal{V}_{\mathrm{A}}$, $\mathcal{V}_{\mathrm{K}}$ and $\mathcal{V}_{\mathrm{B}}$, respectively. The ratio of N_{A}/N is denoted by η_{A}, i.e., the number of application-member sensors to the total number of sensors. Furthermore, let T_{A} and T_{B} denote the number of message communications of sensors in V_{A} and $\mathcal{V}_{\mathrm{B}}$ until the two objectives mentioned in Section II are achieved, respectively, where $T = T_{\mathrm{A}} + T_{\mathrm{B}}$ is the total number of communications. It should be considered in the refined randomized gossip protocol that on the one hand, T_{A} should be small such that the application-member sensors can quickly have the computation output, on the other hand, N_{B} and T_{B} should also be small such that there are only few non-application-member sensors assisting in the communications performing only few number of communications. In this paper, we assume that the non-application-member sensors who assist in the communications can store IHs of the previously transmitted messages in their memory [9]. Let ψ_j denote the number of IHs stored at $v_j \in \mathcal{V}_{\mathrm{B}}$ and the set $\Psi^{v_j} = \{\boldsymbol{I}_l^{v_j} | l = 1, 2, \cdots, \psi_j\}$ contains all the stored IHs at v_j. Communications in randomized gossip are local communications, i.e., between sensors and their neighbor sensors. In the refined protocol, six scenarios are categorized based on the type of the sensor which initiates the communications, referred to as *center*, and the type of its neighbor sensors, as shown in Fig. 1, where filled circles indicate application-member sensors and unfilled circles indicate non-application-member sensors.



(a) Scenario 1 (b) Scenario 2 (c) Scenario 3



(d) Scenario 4 (e) Scenario 5 (f) Scenario 6

Fig. 1: Categorization of 6 different communication scenarios.

A. Protocols for scenarios 1, 2 and 3

In Scenario 1, 2 and 3, the center is an application-member sensor. In Scenario 1 shown in Fig. 1(a), the center as well as all its neighbor sensors are application-member sensors. The protocol for this scenario is the same as we proposed in [9]. In Scenario 2, the center is an application-member sensor, the neighbor sensors consist of both application-member and non-application-member sensors. The center firstly interacts with its application-member neighbor sensors according to [9]. It interacts with its non-application-member neighbor sensors by broadcasting a *request* to all its non-application-member neighbor sensors asking them to assist in the message communications only when the IHs of all application-member sensors are the same, i.e., the data contained in their messages are the same. In Scenario 3, the center has only non-application-member neighbor sensors. When the center initiates the communication, it only broadcasts a *request* to its non-application-member neighbor sensors. The *request* is handled by non-application-member sensors as one of the triggers to initiate the communication when they wake up as the center, the details will be clarified in scenarios 4 to 6.

B. Agency and feedback

We introduce the concept of *agency* to describe the interaction between application-member sensors and non-application-member sensors and the concept of *feedback* to categorize the different interactions. These two concepts help to describe the protocols of scenarios 4, 5 and 6. Since sensors in $\mathcal{V}_\mathrm{B}$ do not generate messages themselves, all the messages they transmit are received from other sensors. In order to reduce T_B for the non-application-member sensors $v_i \in \mathcal{V}_\mathrm{B}$ who have application-member neighbor sensors, i.e., $\mathcal{N}_i^\mathrm{A} \neq \phi$, the times that $v_i \in \mathcal{V}_\mathrm{B}$ receives messages from $\mathcal{N}_i^\mathrm{A}$ should be reduced. To do so, we introduce the term *agency* which uses the assumption of a static network. Intuitively, a non-application-member sensor is an agency when it has no message but have an IH which is the same to the IH of its application-member neighbor sensors. An agency represents its application-member neighbor sensors to interact with other non application-member sensors. Let m_i denote the message at sensor v_i and $m_i = \phi$ indicates that sensor v_i has no message. A sensor $v_i \in \mathcal{V}_\mathrm{B}$ is set as an *agency* of $\mathcal{N}_i^\mathrm{A}$, denoted by $\alpha_i = 1$, under three conditions 1) $m_i = \phi$, 2) $\mathcal{N}_i^\mathrm{A} \neq \phi$ and 3) IHs of all sensors in $\mathcal{N}_i^\mathrm{A}$ are the same. If there is a condition not being fulfilled, the parameter α_i is set to be 0. An agency v_i has no message, however, it has an IH $\boldsymbol{I}_i$ equal to the IH of its application-member sensors, i.e., $\boldsymbol{I}_i = \boldsymbol{I}_j$ for arbitrary $v_j \in \mathcal{N}_i^\mathrm{A}$. In the protocols of scenarios 4 to 6, if a non-application-member sensor $v_i \in \mathcal{V}_\mathrm{B}$ is not an agency $\alpha_i = 0$ nor has a message $m_i = \phi$, we set its IH being null, i.e., $\boldsymbol{I}_i = \mathbf{0}$.

The interaction between non-application-member sensors and their neighbor sensors are based on the *feedbacks*. If a non-application-member sensors $v_i \in \mathcal{V}_\mathrm{B}$ has an IH $\boldsymbol{I}_i \neq \mathbf{0}$, i.e., v_i has a message $m_i \neq \phi$ or v_i is an agency $\alpha_i = 1$, it broadcasts its IH $\boldsymbol{I}_i$ to its neighbors sensors. A neighbor sensor $v_j \in \mathcal{N}_i$ sends *feedbacks* to v_i by comparing the received $\boldsymbol{I}_i$ with its own IH using the function r. The function r takes two IHs $\boldsymbol{I}_i$ and $\boldsymbol{I}_j$ as parameters to enumerate the relations between the corresponding data sets $\mathcal{S}_i = \Theta^{-1}(\boldsymbol{I}_i)$ and $\mathcal{S}_j = \Theta^{-1}(\boldsymbol{I}_j)$ such that [1]

$$r(\boldsymbol{I}_i, \boldsymbol{I}_j) = \begin{cases} 1 & \text{for } \mathcal{S}_i = \mathcal{S}_j \\ 2 & \text{for } \mathcal{S}_i \supset \mathcal{S}_j \\ 3 & \text{for } \mathcal{S}_i \subset \mathcal{S}_j \\ 4 & \text{for all else .} \end{cases}$$

Furthermore, let ε^{v_j} denote a function at non-application-member neighbor sensor $v_j \in \mathcal{N}_i^\mathrm{K}$ of v_i taking $\boldsymbol{I}_i$ as parameter. Function ε^{v_j} compares $\boldsymbol{I}_i$ to all the stored IH $\boldsymbol{I}_l^{v_i} \in \Psi^{v_i}$ at sensor v_j telling whether the data set $\mathcal{S}_i$ at sensor v_i contains new data to the data set $\mathcal{S}_i = \Theta^{-1}(\boldsymbol{I}_l^{v_i})$. If there is new data contained, i.e., $r(\boldsymbol{I}_j, \boldsymbol{I}_l^{v_i}) \in \{2, 4\}$, the results of the function is $\varepsilon^{v_j}(\boldsymbol{I}_i) = 1$, otherwise, the function results is $\varepsilon^{v_i}(\boldsymbol{I}_j) = 0$. Intuitively, the function result of $\varepsilon^{v_j}(\boldsymbol{I}_i)$ tells whether the data set $\mathcal{S}_i$ contains new data that has never been contained in the data set of the message that sensor v_j has transmitted. This function helps non-application-member sensors to omit the communications of the messages that the sensor have transmitted previously. Generally, there are four types of feedbacks from the neighbor sensor v_j to sensor v_i:

T1. if $v_j \in \mathcal{N}_i^\mathrm{A}$ and $r(\boldsymbol{I}_i, \boldsymbol{I}_j) \in \{2, 4\}$;
T2. if $v_j \in \mathcal{N}_i^\mathrm{K}$ with $m_i \neq \phi$ and $r(\boldsymbol{I}_i, \boldsymbol{I}_j) = 2$;
T3. if $\alpha_j = 1$, $r(\boldsymbol{I}_i, \boldsymbol{I}_j) \in \{2, 4\}$ and $\varepsilon^{v_j}(\boldsymbol{I}_i) = 1$;
T4. if $v_j \in \mathcal{N}_i^\mathrm{K}$ with $\mathcal{N}_j^\mathrm{A} = \phi$, $m_j = \phi$, $\boldsymbol{I}_j = \mathbf{0}$ and $\varepsilon^{v_j}(\boldsymbol{I}_i) = 1$.

The T1-feedback is sent to v_i by the neighbor sensor v_j if v_j is an application-member sensor and the data set $\mathcal{S}_i = \Theta^{-1}(\boldsymbol{I}_i)$ contains new data to the current message of sensor v_j. If v_j is a non-application-member sensor with message $m_j \neq \phi$ and the IH comparison results $r(\boldsymbol{I}_i, \boldsymbol{I}_j) = 2$, v_j sends T2-feedback to v_i implying that it can receive the message from v_i and replace its current message with the received one. This is because a non-application-member sensor does not perform computation, but it can simply discard the old message and replace it with the new one. A T3-feedback is sent by an agency, if it detects new data in $\mathcal{S}_i$. A T4-feedback is sent from a non-application-member sensor which does not have a message. This feedback type indicates that the sensor can receive any message as long as $\varepsilon^{v_j}(\boldsymbol{I}_i) = 1$ is fulfilled. A neighbor sensor v_j of sensor v_i will not send any feedback to v_i if non of the conditions in the four types are fulfilled.

In the description of the protocols, we use $\rightarrow$ to indicate the IH transmission and $\Rightarrow$ for the message transmission. If on the left hand side of the arrow there is one sensor, e.g., v_i, and on the right hand side there is one sensor or a set of sensors, it indicates that v_i broadcasts. If on the left hand side there is a set of sensors and on the right hand side there is one sensor, e.g., v_i, it implies that the set of sensors uses a time division mode to transmit messages or indicating headers to v_i. If a sensor $v_i \in \mathcal{V}_\mathrm{A}$ receives a message, let $v_i \uparrow$ indicate that v_i performs data aggregation with the received messages. If $v_i \in \mathcal{V}_\mathrm{B}$ receives a message, $v_i \uparrow$ indicates that it replaces its current message with the one it received if $m_i \neq \phi$ and updates $\boldsymbol{I}_i$ or simply sets $\boldsymbol{I}_i$ to the same as the IH of the received message if $m_i = \phi$. Furthermore, let $\mathcal{N}_i \mapsto v_i$ denote the feedback transmission (including not sending any feedback) from sensors in $\mathcal{N}_i$ to v_i. As mentioned in Scenario 2 and 3, application-member sensors may send a *request* to their non-application-member sensors. Let z_i denote the number of *request*s received by sensor $v_i \in \mathcal{N}_\mathrm{B}$ at the time when sensor v_i wakes up.

---

[1] $A \subset B$ means A is a subset of B, $A \supset B$ means A is a superset of B, $A \cup B$ returns the union of sets A and B.

C. Protocols for scenarios 4, 5 and 6

In Scenarios 4, 5 and 6, the center is a non-application-member sensor. When it wakes up, it initiates the communications under certain conditions. In Scenario 4, the center is $v_i \in \mathcal{V}_\mathrm{B}$ with $\mathcal{N}_i^\mathrm{K} = \phi$. v_i initiates the communications when it wakes up if $z_i = N_i$. The protocol is given in Fig. 2. In this scenario, after v_i receives the IHs from $\mathcal{N}_i$, we use the algorithm proposed in [9] to find the set $\mathcal{P}^1$ which contains the data sets with new data and then further find one IH from all received IHs that contains most new data in the corresponding data set [9].

```
$\mathcal{N}_i \to v_i$;
if $r(\boldsymbol{I}_j, \boldsymbol{I}_k) \neq 1$ for $v_j, v_k \in \mathcal{N}_i, v_j \neq v_k$ then
    $v_i$ applies algorithm in [9] to get $\mathcal{P}^1$;
    $v_i$ finds $v_j$ such that $\Theta^{-1}(\boldsymbol{I}_j) \in \mathcal{P}^1$ contains most new data
    to all other data sets in $\mathcal{P}^1$; $\boldsymbol{I}_i := \boldsymbol{I}_j$;
    $v_j \Rightarrow v_i$;
    $v_i \to \mathcal{N}_i$; $v_i \Rightarrow \mathcal{N}_i$; $v_j \uparrow$ for $v_j \in \mathcal{N}_i$;
    $\Psi^{v_i} := \Psi^{v_i} \cup \boldsymbol{I}_j$; $m_i := \phi$; $z_i := 0$;
else
    $z_i := 0$;
end if
```

Fig. 2: Protocol for Scenario 4

In Scenario 5, the center is $v_i \in \mathcal{V}_\mathrm{B}$ where $\mathcal{N}_i^\mathrm{A} \neq \phi$ and $\mathcal{N}_i^\mathrm{K} \neq \phi$. v_i initiates the communications if $z_i = N_i^\mathrm{A}$ or $m_i \neq \phi$. In this scenario, if the center v_i has a message $m_i \neq \phi$, it broadcasts the message if it receives any T1, T2 or T3 feedbacks. If $m_i = \phi$, v_i checks whether it is an agency sensor. If $\alpha_i = 0$, it helps the message communications among $\mathcal{N}_i^\mathrm{A}$ using the protocol in Fig. 2, otherwise it forwards the message from $\mathcal{N}_i^\mathrm{A}$ to sensors in $\mathcal{N}_i^\mathrm{K}$. The protocol is given in Fig. 3.

In Scenario 6, the center is $v_i \in \mathcal{V}_i^\mathrm{B}$ and $\mathcal{N}_i^\mathrm{A} = \phi$. In some situations, it may happen that v_i as well as all sensors in $\mathcal{N}_i$ have messages with new data. In this case, the message exchange is impossible since non-application-member sensors cannot perform computations. To overcome this situation, the center v_i may use its memory as a *stack* to store only one message. Let m_i^S denote the message in the stack of sensor v_i, if $m_i^\mathrm{S} = \phi$, the stack is free. After the message is stored in the stack, v_i will send a T4-feedback when it receives IH from other sensors. Therefore, in this scenario, v_i initiates the communications if 1) $m_i \neq \phi$ or 2) there is message stored in the stack of v_i. The protocol of this scenario is given in Fig. 4.

```
if $m_i \neq \phi$ then
    $v_i \to \mathcal{N}_i$;
    $\mathcal{N}_i \mapsto v_i$;
    if $v_i$ receives any T1, T2 or T3 feedbacks then
        $v_i \Rightarrow \mathcal{N}_i$; $v_j \uparrow$ for $v_j \in \mathcal{N}_i$;
    end if
    $\Psi^{v_i} := \Psi^{v_i} \cup \boldsymbol{I}_i$; $m_i := \phi$; $z_i := 0$;
else
    $\mathcal{N}_i^\mathrm{A} \to v_i$;
    if $r(\boldsymbol{I}_j, \boldsymbol{I}_k) \neq 1$ for $v_j, v_k \in \mathcal{N}_i^\mathrm{A}, v_j \neq v_k$ then
        Repeat Protocol for Scenario 4 in Fig. 2.
    else
        $\alpha_i := 1$
        $v_i \to \mathcal{N}_i^\mathrm{K}$;
        $\mathcal{N}_i^\mathrm{K} \mapsto v_i$;
        if $v_i$ receives any T1, T2 or T3 feedbacks then
            $v_j \Rightarrow v_i$, with an arbitrary $v_j \in \mathcal{N}_i^\mathrm{A}$.
            $v_i \Rightarrow \mathcal{N}_i^\mathrm{K}$; $v_l \uparrow$ for $v_l \in \mathcal{N}_i^\mathrm{K}$ which sent T1,T2 or T3
            feedback;
        else
            if any T4 feedback from $\mathcal{N}_i^\mathrm{K}$ then
                $v_j \Rightarrow v_i$, with an arbitrary $v_j \in \mathcal{N}_i^\mathrm{A}$.
                $v_i$ randomly chooses one $v_j$ which sent T4 feedback;
                $v_i \Rightarrow v_j$; $v_j \uparrow$;
            end if
        end if
        $\Psi^{v_i} := \Psi^{v_i} \cup \boldsymbol{I}_i$; $m_i := \phi$; $z_i := 0$;
    end if
end if
```

Fig. 3: Protocol for Scenario 5

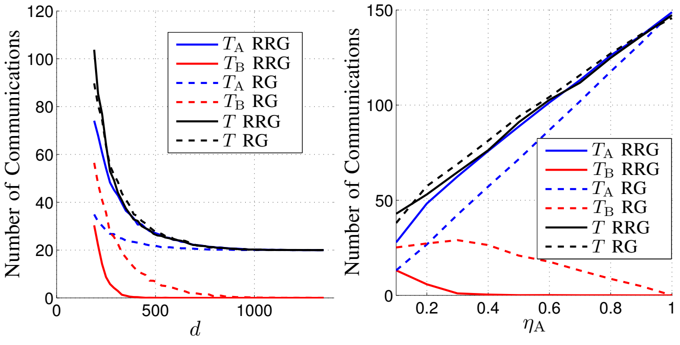
```
if $m_i = \phi$ and $m_i^\mathrm{S} \neq \phi$ then
    $m_i := m_i^\mathrm{S}$; $m_i^\mathrm{S} := \phi$; set $\boldsymbol{I}_i$ as the IH of $m_i^\mathrm{S}$;
end if
$v_i \to \mathcal{N}_i$;
$\mathcal{N}_i \mapsto v_i$;
if $v_i$ receives no feedback from $\mathcal{N}_i^\mathrm{K}$ then
    switch $m_i$ and $m_i^\mathrm{S}$;
else
    if $v_i$ receives any T2 or T3 feedback then
        $v_i \Rightarrow \mathcal{N}_i^\mathrm{K}$; $v_j \uparrow$ for $v_j \in \mathcal{N}_i^\mathrm{K}$ which sent T2 or T3
        feedback;
        $\Psi^{v_i} = \Psi^{v_i} \cup \boldsymbol{I}_i$; $m_i := m_i^\mathrm{S}$; $m_i^\mathrm{S} := \phi$; Update $\boldsymbol{I}_i$;
    else
        if $v_i$ receives any T4 feedback then
            $v_i$ randomly choose one $v_j$ which sent T4 feedback;
            $v_i \Rightarrow v_j$; $v_j \uparrow$;
            $\Psi^{v_i} = \Psi^{v_i} \cup \boldsymbol{I}_i$; $m_i := m_i^\mathrm{S}$; $m_i^\mathrm{S} := \phi$; Update $\boldsymbol{I}_i$;
        end if
    end if
end if
```

Fig. 4: Protocol for Scenario 6

V. SIMULATIONS

In the simulations, we consider a WSN with $N = 100$ sensors randomly deployed in a 1000-by-1000 squared area. The Refined Randomized Gossip (RRG) is compared to the randomized gossip (RG) considered in [9]. In the RG, the same sensors as the application-member sensors in the RRG generate measurement data, the others generate no data. In contrast to RRG, all non-application-member sensors in the RG behave like application-member sensors, i.e., using the protocol defined in Scenario 1. The objectives of RG are 1) to compute the same divisible function as RRG and 2) to let the application-member sensors know the function output. For both approaches, the communications stop when all application-member sensors have the aggregated output function.

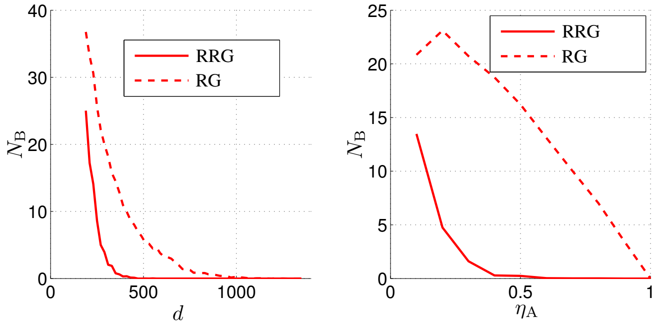
In Fig. 5, we consider the number of communications T_A, T_B and T with respect to the communication range d as well as the ratio η_A of the application-member sensors in the network. In Fig. 5a with $\eta_\mathrm{A} = 0.2$, we vary d which takes



(a) (b)

Fig. 5: (a): Number of communications vs. d. (b): Number of communications vs. η_A

the value starting from the minimum value (approximately 200) that leads to a connected network to the value with which all sensors are connected to all the other sensors (diagonal distance of the squared area). With almost the same total number T, the number T_B of communications for the non-application-member sensors using RRG are significantly reduced in comparison to that using RG. However, the price payed for such improvement is a higher number T_A of communications by application-member sensors. When increasing the communication range, a sensor has more neighbor sensors. Therefore, the probability that application-member sensors can directly communicate increases. This results in a decreased number of communications for both application-member sensors and non-application-member sensors. The Fig. 5b compares the number of communications for different η_A with a fixed $d = 270$. RRG outperforms RG in terms of the number of communications T_B performed by non-application-member sensors. In Fig. 6a and 6b, the number of involved non-application-member sensors N_B is depicted versus d and the ratio η_A, respectively. The results show the reduction of the involvement of non-application-member sensors using the RRG in comparison to the RG protocols.



(a) (b)

Fig. 6: (a): N_B vs. d. (b): N_B vs. η_A

VI. CONCLUSION

In this paper, we propose a refined randomized gossip protocol for wireless sensor networks where only part of the sensors are involved in the application, i.e. only a subset of the sensors generates measurement data and is interested in the function output taking the measurement data as parameters. Sensors are categorized into application-member sensors and non-application member sensors depending on their involvement in the application. Non-application-member sensors need to assist in the communications between application-member sensors. Our newly proposed refined protocol minimizes the number of involved non-application-member sensors and the number of their communications. Depending on the type of neighbor sensors, communication protocols for six different scenarios are discussed. Performance evaluations show the reduction of the number of communications performed by the non-application-member sensors as well as the number of non-application-member sensors that are involved in the proposed protocols in comparison to the approach where all sensors communicate with each other as application-member sensors considered in our previous work.

ACKNOWLEDGMENT

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] Y. Lin, S. Chang, and H. Sun, “CDAMA: Concealed data aggregation scheme for multiple applications in wireless sensor networks,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 7, pp. 1471 – 1483, 2013.

[2] Z. Chen, A. Kuehne, A. Klein, A. Loch, M. Hollick, and J. Widmer, “Two-way relaying for multiple applications in wireless sensor networks,” in *Proc. International ITG Workshop on Smart Antennas*, Mar 2012.

[3] M. Shazly, E. Elmallah, and J. Harms, “Balancing area coverage in partitioned wireless sensor networks,” in *Proc. Wireless Communications and Networking Conference*, April 2012.

[4] 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.

[5] 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.

[6] 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.

[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] ——, “Multi-hop coordination in gossiping-based wireless sensor networks,” in *Proc. 10th International Symposium on Wireless Communication Systems*, 2013.

[9] ——, “Improved bias cancellation and header communication for random gossiping-based static wireless sensor networks,” in *Proc. 18th International ITG Workshop on Smart Antennas*, 2014.

[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] 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.