International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2014): 5.611

Fault Detection in Wireless Sensor Network Depending On Health State Using Distributed Approach

Neha Belwalkar, Prof. V. S. Deshmukh

Department of Computer Networking, SKNCOE, Pune
Savitribai Phule Pune University, Pune

Abstract: ***This paper proposed a distributed localized faulty sensor detection algorithm where each sensor identifies its own status to be either "good" or "faulty" and the claim is then supported or reverted by its neighbors as they also evaluate the node behavior.***

Keywords: Fault Detection, WSN, Health State, Distributed Approach

1. Introduction

In a common Wireless Sensor Network architecture, the measurement nodes are deployed to calculate measurements such as temperature, voltage, heat, or even dissolved oxygen. The nodes are part of a wireless sensor network administered by the gateway that governs network aspects such as client authentication and data security [14]. The gateway collects the measurement data from each and every node and sends it through a wired connection, typically Ethernet, to a host controller. 1.5.1 Networking Topologies We can use several network topologies to coordinate the Wireless sensor network gateway, end nodes, and router nodes. Router nodes are much similar to end nodes in that they can store measurement data, but they also can be used to pass along measurement data from other nodes. The first, and most basic topology, is the star topology, in which each node maintains a single, direct communication link with the gateway. This topology is very simple but restricts the overall distance that our network can achieve. To increase the distance that a network can cover, you could implement a cluster, or tree, topology. In this more complex architecture scenario, each node still uses only one communication path to the gateway but can use other nodes to route its information along that path. This topology suffers from a typical problem, however. If a router node goes down, all the nodes which depend on that router node also lose their communication links to the gateway. The mesh network topology reduces this issue by extensively using redundant communication paths to increase reliability of the system. In a mesh network, nodes maintain multiple communication links back to the gateway, so that if a router node goes down or does not work properly, the network automatically reroutes the data through a different sets of path. The mesh topology, although very reliable, suffers from an wide increase in network latency because data must make multiples of hops before successfully arriving at the gateway 14.

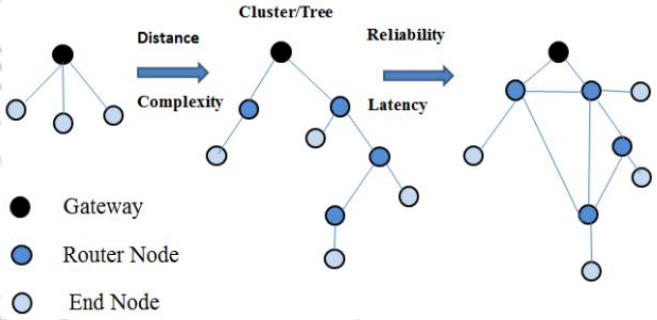

Figure 1: WSN Network Topologies

2. Literature Survey

In this section, we like to give a brief review of the different schemes that we have studied for wireless sensor networks.

2.1 A Self-Managing Fault Management Mechanism for WSN

In this approach a new fault management mechanism was proposed to deal with fault detection and recovery. It proposes a hierarchical structure to properly distribute fault management tasks among sensor nodes by heavily introducing more self-managing functions. The proposed failure detection and recovery algorithms have been compared with some existing related algorithm and proven to be more energy efficient. The proposed fault management mechanism can be divided into two phases: o Fault detection and diagnosis o Fault recovery.

2.2 Fault Detection and Diagnosis

Detection of faulty sensor nodes can be achieved by two mechanisms i.e. self-detection (or passive-detection) and active-detection. In self-detection, sensor nodes are required to periodically monitor their residual energy, and identify the potential failure. In this scheme, we consider the battery depletion as a main cause of node sudden death. A node is termed as failing when its energy drops below the threshold value. When a common node is failing due to energy depletion, it sends a message to its cell manager that it is

going to sleep mode due to energy below the threshold value. This requires no recovery steps. Self-detection is considered as a local computational process of sensor nodes, and requires less in-network communication to conserve the node energy. In addition, it also reduces the response delay of the management system towards the potential failure of sensor nodes. To efficiently detect the node sudden death, our fault management system employed an active detection mode. In this approach, the message of updating the node residual battery is applied to track the existence of sensor nodes. In active detection, cell manager asks its cell members on regular basis to send their updates. Such as the cell manager sends "get" messages to the associated common nodes on regular basis and in return nodes send their updates. This is called in-cell update cycle. The update_msg consists of node ID, energy and location information. As shown in figure 2, exchange of update messages takes place between cell manager and its cell members. If the cell manager does not receive an update from any node then it sends an instant message to the node acquiring about its status. If cell manager does not receive the acknowledgement in a given time, it then declares the node faulty and passes this information to the remaining nodes in the cell. Cell managers only concentrate on its cell members and only inform the group manager for further assistant if the network performance of its small region has been in a critical level.

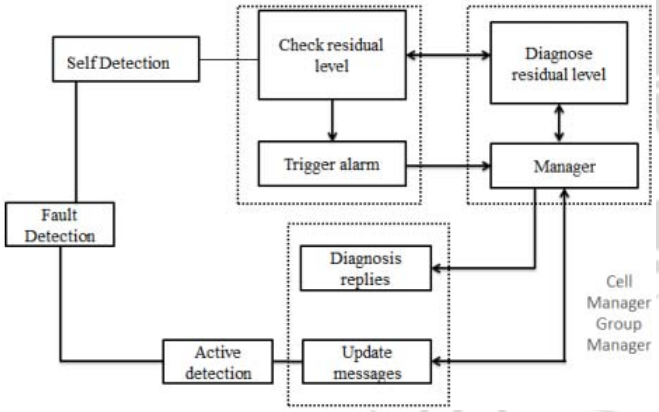

Figure 2: Fault Detection and Diagnosis Process

A cell manager also employs the self-detection approach and regularly monitors its residual energy status. All sensor nodes start with the same residual energy. After going through various transmissions, the node energy decreases. If the node energy becomes less than or equal to 20% of battery life, the node is ranked as low energy node and becomes liable to put to sleep. If the node energy is greater or equal to 50% of the battery life, it is ranked as high and becomes the promising candidate for the cell manager. Thus, if a cell manager residual energy becomes less than or equal to 20% of battery life, it then triggers the alarm and notifies its cell members and the group manager of its low energy status and appoints a new cell manager to replace it.

Every cell manager sends health status information to its group manager. This is called out-cell update cycle and are less frequent than in-cell update cycle. If a group manager does not hear from a particular cell manager during out-cell update cycle, it then sends a quick reminder to the cell manager and enquires about its status. If the group manager does not hear from the same cell manager again during second update cycle, it then declares the cell manager faulty and informs its cell members [20]. This approach is used to detect the sudden death of a cell manager. Group manager also monitor its health status regularly and respond when its residual energy drops below the threshold value. It notifies its cell members and neighboring group managers of its low energy status and an indication to appoint a new group manager. Sudden death of a group manager can be detected by the base station. If the bases station does not receive any traffic from a particular group manager, it then consults the group manager and asks for its current status. If the base station does not receive any acknowledgement, it then considers the group manager faulty (sudden death) and propagates this information to its cell managers. The base station 18 primarily focuses on the existence of the group managers from their sudden death. Meanwhile, the group managers and cell managers take most parts in passive and active detection in the network.

2.3 Fault Recovery

After nodes failure detection (as a result of self-detection or active detection), sleeping nodes can be awaked to cover the required cell density or mobile nodes can be moved to fill the coverage hole. A cell manager also appoints a secondary cell manager within its cell to acts as a backup cell manager. Cell manager and secondary cell manager are known to their cell members. If the cell manager energy drops below the threshold value (i.e. less than or equal to 20% of battery life), it then sends a message to its cell members including secondary cell manager. It also informs its group manager of its residual energy status and about the candidate secondary cell manager. This is an indication for secondary cell manager to stand up as a new cell manager and the existing cell manager becomes common node and goes to a low computational mode. Common nodes will automatically start treating the secondary cell manager as their new cell manager and the new cell manager upon receiving updates from its cell members; choose a new secondary cell manager. The failure recovery mechanisms are performed locally by each cell. In Figure 3, let us assume that cell 1 cell manager is failing due to energy depletion and node 3 is chosen as secondary cell manager. Cell manager will send a message to node 1, 2, 3 and 4 and this will initiate the recovery mechanism by invoking node 3 to stand up as a new cell manager.

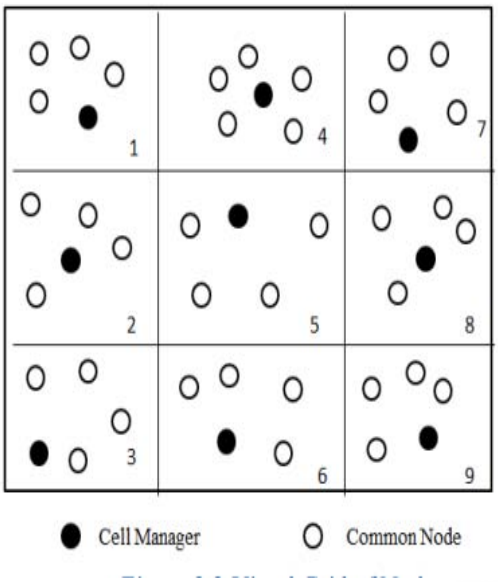

Figure 3: Virtual Grid of Nodes

In a scenario, where the residual battery energy of a particular cell manager is not sufficient enough to support its management role, and the secondary cell manager also does not have sufficient energy to replace its cell manager. Thus, common nodes exchange energy messages 19 within the cell to appoint a new cell manager with residual energy greater or equal to 50% of battery life. In addition, if there is no candidate node within the cell that has sufficient energy to replace the cell manager. The event cell manager sends a request to its group manager to merge the remaining nodes with the neighboring cells.

When a group manager detects the sudden death of a cell manager, it then informs the cell members of that faulty cell manager (including the secondary cell manager). This is an indication for the secondary cell manager to start acting as a new cell manager. A group manager also maintains a backup node within the group to replace it when required. If the group manager residual energy drops below the threshold value (i.e. greater or equal to 50% of battery life), it may downgrade itself to a common node or enter into a sleep mode, and notify its backup node to replace it. The information of this change is propagated to neighboring group managers and cell managers within the group. As a result of group manager sudden death, the backup node will receive a message from the base station to start acting as the new group manager. If the backup node does not have enough energy to replace the group manager, cell managers within a group co-ordinate to appoint a new group manager for themselves based on residual energy. Each cell maintains its health status in terms of energy. It can be High, Medium or Low. These health statuses are then sent out to their associate group managers periodically during outcell update cycle. Upon receiving these health statuses, group manager predict and avoid future faults. For example; if a cell has health status high then group manager always recommends that cell for any operation or routing but if the health status is medium then group manager will occasionally recommend it for any operation. Health status Low means that the cell has insufficient energy and should be avoided for any operation. Therefore, a group manager can easily avoid using cells with low health status or alternatively, instruct the low health status cell to join the neighboring cell. Consider Figure 3, let cell 4 manager is a group manager and it receives health status updates from cell 1, 2 and 3. Cell 2 sends a health status low to its group manager, which alert group manager about the energy status of cell 2.

3. Distributed Fault Detection in WSN

3.1 Definition

n: total number of sensors;

p: probability of failure of a sensor;

k: number of neighbor sensors;

S: set of all the sensors;

$N(S_i)$: set of the neighbors of S_i;

x_i: measurement of S_i;

d_{ij}^t: measurement difference between S_i and S_j at time t,

$d_{ij}^t = x_i^t - x_j^t$;

$\Delta t_l = t_{l+1} - t_l$;

$\Delta d_{ij}^{\Delta t_l}$: measurement difference between S_i and S_j from time t_l to t_{l+1}, $\Delta_{ij}^{\Delta t_l} = d_{ij}^{t_{l+1}} - d_{ij}^{t_l} = (x_i^{t_{l+1}} - x_j^{t_{l+1}}) - (x_i^{t_l} - x_j^{t_l})$

c_{ij}: test between S_i and S_j, $c_{ij} \in \{0, 1\}$, $c_{ij} = c_{ji}$;

$\theta 1$ and $\theta 2$: two predefined threshold values;

T_i: tendency value of a sensor, $T_i \in \{LG, LF, GD, FT\}$;

3.2 Existing Algorithm

STEP 1

Each sensor S_i, set $c_{ij} = 0$ and compute d_{ij}^t;

IF $|d_{ij}^t| > \theta 1$ THEN

Calculate $\Delta l_{ij}^{\Delta t_l}$;

IF $|\Delta d_{ij}^{\Delta t_l}| > \theta 2$ THEN $c_{ji} = 1$;

STEP 2

IF $\sum_{S_j \in N(S_i)} c_{ij} < \lceil |N(S_i)|/2 \rceil$ where $|N(S_i)|$ is the number of the S_i's neighboring nodes THEN

$T_i = LG$;

ELSE $T_i = LF$;

Communicate T_i to neighbors;

STEP 3

$$IF \sum_{S_j \in N(S_i) \text{ and } T_j=LG} (1 - 2c_{ij}) \geq \lceil |N(S_i)|/2 \rceil$$

$$THEN$$

$$T_i = GD;$$

$$Communicate\ T_i\ to\ neighbors;$$

STEP 4

$$FOR\ i = 1\ to\ n$$

$$IF\ T_i = LG\ or\ T_i = LF\ \ THEN$$

$$IF\ T_j = GD\ \ \forall S_j \in N(S_i)\ \ THEN$$

$$IF\ c_{ij} = 0\ \ THEN$$

$$T_i = GD;$$

$$ELSE\ T_i = FT;$$

$$ELSE\ repeat$$

$$Communicate\ T_i\ to\ neighbors;$$

STEP 5

$$FOR\ each\ S_i,\ IF\ T_j = T_h = GD$$

$$\forall S_j,\ S_h \in N(S_i),\ where\ j \neq h,$$

$$and\ IF\ c_{ji} \neq c_{hi}\ \ THEN$$

$$IF\ T_i = LG\ (or\ LF)\ \ THEN$$

$$T_i = GD\ (or\ FT)$$

Sensors are considered as neighboring sensors if they are within the transmission range of each other. Each node regularly sends its measured value to all its neighbors. We are interested in the history data if more than half of the sensor's neighbors have a significantly different value from it. We can use this $\Delta d^{\Delta ij}_{ij}$ to find if the current measurement is different from previous measurement. If the measurements change over the time significantly, it is more likely the sensor is faulty.

A test result cij is generated by sensor Si based on its neighbor Sj 's measurements using two variables, d^{t}_{ij} and $\Delta d^{\Delta ij}_{ij}$, and two predefined threshold value θ1 and θ 2 . If a sensor is faulty, it can generate arbitrary measurements. If cij is 0, most likely either both Si and Sj are good or both are faulty. Otherwise, if cij is 1, Si and Sj are most likely in different status. Sensors can be either LG or LF, determined by using test value from its neighboring sensors. Each sensor sends its tendency value to all its neighbors. The number of the LG sensors with coincident test results determines whether the sensors are GD or FT.

3.3 An Example

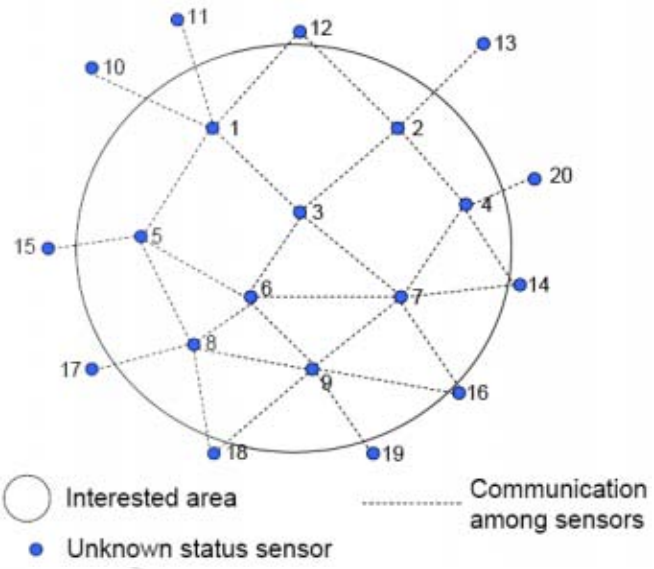

Figure 4: A partial set of sensor nodes in a WSN with faulty sensors

In this section, we present an example to illustrate our algorithm. Fig.4 shows a partial set of sensor nodes in a wireless sensor network with some faulty nodes. Nodes S1 − S9 inside the circle area are the nodes which we are interested in. If the two nodes are neighbors, they are connected by dotted line. Communication between nodes outside the circle is not shown in the figure. Each node inside the interested area is tested by its neighbors. Test results are either 0 or 1 depending upon the measurement difference and threshold value θ . Tendency value Ti is finalized at the third iteration.

4. Proposed Mechanism

4.1 Network model and Fault model

We assume that sensors are randomly deployed in the interested area which is very dense and all the sensors have a common transmission range. The dark circles in the figure represent faulty sensors and the gray circles are good sensors. There might be a failure occurring in a certain area as illustrated in the figure 2 All sensors in this area go out of service. As we are depending on majority voting among the sensors, we assume that each sensor node has at least 3 neighboring nodes. Because a large amount of sensors are deployed into the interested area to form a wireless network, this condition can be easily obtained. Each sensor node is able to locate its neighbors within its transmission range via a broadcast/ acknowledge protocol. Faults can occur at different levels of the sensor network, such as system software, hardware, physical layer, and middleware. In this mechanism, we focus on hardware level faults by assuming all system software as well as the application software is always fault tolerant. We can categorize the hardware components of sensor nodes into two groups. The first group of hardware level components consists of a storage subsystem, computation engine and power supply infrastructure. The second groups of components are sensors and actuators. The second group is most prone to

malfunctioning. We only consider the sensor faults which occur in the second group. Sensor nodes are still capable of receiving, sending, and processing when they are faulty in the algorithm.

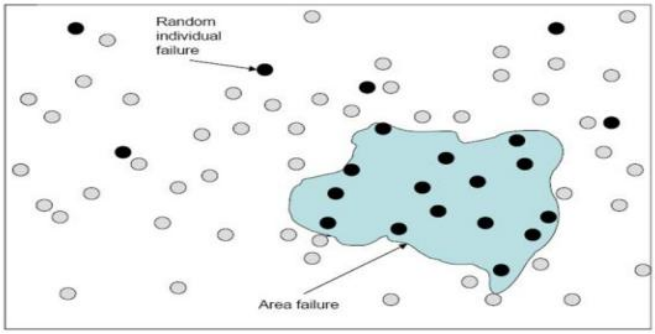

Figure 4: Sensor nodes randomly deployed over an area

4.2 Definition

n: total number of sensors;

p: probability of failure of a sensor;

k: number of neighbor sensors;

S: set of all the sensors;

$N(S_i)$: set of the neighbors of S_i;

x_i: measurement of S_i;

d_{ij}^t: measurement difference between S_i and S_j at time t,

$d_{ij}^t = x_i^t - x_j^t$;

$\Delta t_l = t_{l+1} - t_l$;

$\Delta d_{ij}^{\Delta t_l}$: measurement difference between S_i and S_j from time t_l to t_{l+1}, $\Delta_{ij}^{\Delta t_l} = d_{ij}^{t_{l+1}} - d_{ij}^{t_l} = (x_i^{t_{l+1}} - x_j^{t_{l+1}}) - (x_i^{t_l} - x_j^{t_l})$

c_{ij}: test between S_i and S_j, $c_{ij} \in \{0, 1\}$, $c_{ij} = c_{ji}$;

$\theta 1$ and $\theta 2$: two predefined threshold values;

T_i: tendency value of a sensor, $T_i \in \{LG, LF, GD, FT\}$;

Sensors are considered as neighboring sensors if they are within the transmission range of each other. Each node regularly sends its measured value to all its neighbors. We are interested in the history data if more than half of the sensor's neighbors have a significantly different value from it. We can find the current measurement is different from previous measurement. If the measurements change over the time significantly, it is more likely the sensor is faulty [3].

A test result Cij is generated by sensor Si based on its neighbor Sj's measurements using two variables and two predefined threshold value. If a sensor is faulty, it can generate arbitrary measurements. If Cij is 0, most likely either both Si and Sj are good or both are faulty. Otherwise, if Cij is 1, Si and Sj are most likely in different status.

5. Issues in the Existing Algorithm

From the realization of DFD node fault detection scheme, for a normal node Snormal, if the number of its neighbor nodes having initial detection status of LG is less than $\lceil |N(Snormal)| / 2 \rceil$, then Snormal is misdiagnosed as faulty, thus reducing the fault detection accuracy. The conditions of detecting the normal node as "normal" are too harsh in DFD node fault detection scheme. Besides, the node fault accuracy of DFD scheme will decrease rapidly when there are not many neighbors of the nodes to be detected or the node's failure ratio of network is high. 29 The improved DFD node fault detection scheme proposed in this project changes the detection criterion of DFD scheme as follows: For any node Si and the nodes in N(Si) whose initial detection status is LG, if the nodes whose test result with Si is 0 are not less than the nodes whose test result is 1, then the status of Si is normal (GD), otherwise, the status of Si is faulty (FT).

6. Proposed Algorithm

STEP 1
Each sensor Si and any sensor Sj ∈ N(Si) set cij = 0 and compute d^t_{ij}
IF $|d^t_{ij}| > \theta$ THEN cij = 1 and turn to the next node in N(Si);
IF $|d^t_{ij}| < \theta$l Calculate $\Delta a^{\Delta ti}_{ij}$;
IF $|d^t_{ij}| > \theta 2$ then cij = 1 and turn to the next node in N(Si) ;
repeat above steps until the test results of each node in N(Si) with Si are all obtained;

STEP 2
IF
$\sum_{S_j \in N(S_i)} c_{ij} < \lceil |N(S_i)|/2 \rceil$, where $|N(S_i)|$ is
the number of the Si 's neighboring nodes
THEN Ti = LG;
ELSE Ti = LF;
Communicate T to neighbors;

STEP 3
IF
$\sum_{S_j \in N(S_i) \text{ and } T_j = LG} C_{ij} < \lceil |N(S_i)_{Tj = LG}|/2 \rceil$
THEN Ti = GD;
ELSE
THEN Ti = FT;
Communicate Ti to neighbors;

STEP 4
If there are no neighbor nodes of Si
Whose initial detection status is LG, and if the initial detection status Ti of Si is LG, then set the
status of Si as normal (GD), otherwise as fault(FT);

STEP 5
Check whether detection of the status of all
nodes in network is completed or not. If it has been completed,
then exit. Otherwise, repeat steps of (1), (2), (3) and (4).

7. Conclusion

We proposed a distributed localized faulty sensor detection algorithm where each sensor identifies its own status to be either ”good” or ”faulty” and the claim is then supported or reverted by its neighbors as they also evaluate the node behavior

References

[1] R. Verdone, D. Dardari, G. Mazzini, and A. Conti, Wireless sensor and actuator networks: technologies, analysis and design. Academic Press, 2010.
[2] T. Quek, D. Dardari, and M. Win, “Energy efficiency of dense wireless sensor networks: To cooperate or not to cooperate,” IEEE Journal on Selected Areas in Communications, vol. 25, no. 2, pp. 459–470, February 2007.
[3] J. Chen, S. Kher, and A. Somani, “Distributed fault detection of wireless sensor networks,” in Proc Workshop DIWANS, New York, NY, USA, 2006, pp. 65 – 72.
[4] M. Ding, D. Chen, K. Xing, and X. Cheng, “Localized fault-tolerant event boundary detection in sensor networks,” in Proc IEEE INFOCOM, Miami, FL, USA, 2005, pp. 902 – 913.
[5] J.-L. Gao, Y.-J. Xu, and X.-W. Li, “Weighted-median based distributed fault detection for wireless sensor networks,” Journal of Software, vol. 18, no. 5, pp. 1208 – 1217, 2007.
[6] S. Ji, S.-F. Yuan, T.-H. Ma, and C. Tan, “Distributed fault detection forwireless sensor based on weighted average,” in Proc NSWCTC, Wuhan,China, 2010, pp. 57 – 60.
[7] M.-H. Lee and Y.-H. Choi, “Fault detection of wireless sensor networks,” Computer Communications, vol. 31, no. 14, pp. 3469–3475,2008.
[8] M. Cheraghchi, A. Hormati, A. Karbasi, and M. Vetterli, “Group testing with probabilistic tests: theory, design and application,” IEEE Trans. Inf. Theory, vol. 57, no. 10, pp. 7057 – 7067, 2011.
[9] T. Tosic, N. Thomos, and P. Frossard, “Distributed sensor failure detection in sensor networks,” Signal Processing, vol. 93, no. 2, pp. 399–410, 2013.
[10] M. Milanese, J. Norton, H. Piet-Lahanier, and E. Walter, Eds., Bounding Approaches to System Identification. New York, NY: Plenum Press, 1996.
[11] L. Jaulin, M. Kieffer, O. Didrit, and E.Walter, Applied Interval Analysis. London: Springer-Verlag, 2001.
[12] M. Haenggi, Stochastic geometry for wireless networks. Cambridge University Press, 2012