

Connectivity-Based Self-Localization in WSNs

Jozef KENYERES^{1,2}, Martin KENYERES², Markus RUPP¹, Peter FARKAŠ^{2,3}

¹Institute of Telecommunications, Vienna University of Technology, Gusshausstr. 25/389, E389 1040 Vienna, Austria

²Institute of Telecommunications, Slovak University of Technology, Ilkovičova 3, 812 19, Bratislava, Slovakia

³Faculty of Informatics, Pan-European University, Nevädzová 5, 821 01 Bratislava, Slovakia

jozef.kenyeres@tuwien.ac.at, markus.rupp@tuwien.ac.at

Abstract. *Efficient localization methods are among the major challenges in wireless sensor networks today. In this paper, we present our so-called connectivity based approach, i.e. based on local connectivity information, to tackle this problem. At first the method fragments the network into larger groups labeled as packs. Based on the mutual connectivity relations with their surrounding packs, we identify border nodes as well as the central node. As this first approach requires some a-priori knowledge on the network topology, we also present a novel segment-based fragmentation method to estimate the central pack of the network as well as detecting so-called corner packs without any a-priori knowledge. Based on these detected points, the network is fragmented into a set of even larger elements, so-called segments built on top of the packs, supporting even more localization information as they all reach the central node.*

Keywords

WSN, distributed algorithms, border nodes, virtual coordinates.

1. Introduction

Wireless sensor networks (WSNs) attracted much attention by both academia and industry in the last years. As presented in [1], WSNs are facing a vast number of serious challenges; those related to the energy constraints being the most crucial ones. This emphasizes the need of viable and energy efficient solutions, together with other important challenges that arise in a possible large number of nodes. The authors in [2] concluded that for many areas of WSNs such solutions exist and are already standardized such as [3], [4], [5] and [6]. On the other hand, a global consensus about a routing protocol suitable for WSNs is lacking. The authors in [2] present a detailed survey of many different approaches. They start with out-of date solutions such as *flooding* and *cluster-based hierarchical* protocols. Then, they continue with *geographical* routing protocols following the approach in [7]. In these protocols, their necessity of the node's lo-

cation awareness is a weakness. To overcome it, so called virtual coordinates emerged. They are intended to be used instead of the physical coordinates to define the node's location. The protocol class related to virtual coordinates is labeled as *self-organizing coordinate systems* in [2].

Self-organizing coordinate systems are considered as the current state of the art. Nevertheless, many challenges still remain; the most obvious one being the way, in which the nodes obtain their coordinates. Currently, several solutions exist. A straightforward solution assumes the presence of anchor points in the network. They are used as a location reference for the rest of the nodes, which obtain their location by a local measurement. There are various methods how to accomplish this task, for example Received Signal Strength (RSS) and Time Of Arrival (TOA) as in [8] or Time of Flight (ToF) from [9]. Also, a multilateration method similar to GPS-based systems is presented in [10]. The major drawback of these methods presented in [2] is that these derivatives of the physical coordinates are not reflecting the network topology. Also, the precision of these methods is questionable. An other approach is based on the assumption that the traffic destinations are known in WSNs (sink nodes or base stations). Thus, they serve as reference points. The nodes are able to determine their location, based on their distances from these points as presented in [11], [12] and [13]. As there are no natural reference points expected in the network, they have to be picked. This should be performed randomly as in [14], or on the network's border as in [18], [15], [16]. Another approach is the so-called centroid transformation presented in [17].

In our work, we target this problem of node self-localization. We introduce the concept of connectivity based localization with the definition of the most important terms in Section 2 and describe more details of our algorithm in Section 3 based on our earlier work [19], [20]. Section 4 gained results of such algorithm obtained by simulations. In this paper, we present a significant extension of our algorithm in Section 5. Here, we aim to provide every node in the network with location data and to generalize our algorithm in order to gain a valid result regardless of the network topology. We analyze the improved results in Section 6 and conclude the paper in Section 7. All new and unpublished results are thus presented in Sections 5 and 6.

2. Connectivity-Based Localization

In our work presented in [19] and [20], we introduced a novel localization concept, suitable for WSN nodes. We made several assumptions for a WSN:

- A WSN $\mathcal{N} = \{n_1, n_2, \dots, n_{|\mathcal{N}|}\}$ consists of a large number of the nodes (typically $|\mathcal{N}| > 100$). We denote the size of the WSN by $|\mathcal{N}|$.
- The nodes $n_i; i = 1, 2, \dots, |\mathcal{N}|$ are randomly located within a plane and their location is defined by an ordered pair of two coordinates (x_i, y_i) . We assume that the nodes are located within a squared plane consisting of $M \times M$ discrete points. Their location within the plane is defined by an ordered pair of two coordinates $(x, y), x \in \{1, 2, \dots, M\} = \mathcal{M}; y \in \mathcal{M}$. The position $(x_i, y_i) \in \mathcal{M}^2$ of every node n_i is unique, i.e., $\nexists n_i \vee n_j | (x_i, y_i) = (x_j, y_j)$.
- The transmission area of every node is defined as a circle with the node located in its center and with a defined radius r , equal to all nodes. Every two nodes with a mutual distance smaller than such radius are considered as adjacent nodes, reflected by the elements $a_{ij} \in \{0, 1\}$ of adjacency matrix \mathbf{A} with $a_{ii} = 0$. Take, for example, node n_i ; the center of its area is located in the point (x_i, y_i) and it covers an area of size πr^2 . Consequently, every n_j with Euclidean distance E to n_i smaller or equal to r becomes its adjacent node. This can be stated as follows:

$$a_{ij} = 1 | E(n_i, n_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \leq r. \quad (1)$$

Consequently, every network can be described by an ordered triplet $(|\mathcal{N}|, M, r)$.

- Every message send by a node is correctly received by all its adjacent nodes. In order to describe the network's connectivity, we define parameter ξ . It defines an expected average number of adjacent nodes of every node n_i and is stated as follows:

$$\xi = \frac{|\mathcal{N}|}{M^2} \pi r^2, \quad (2)$$

thus given the triplet $(|\mathcal{N}|, M, r)$, the network's connectivity is defined. The ratio $\frac{|\mathcal{N}|}{M^2}$ can be interpreted as the average number of nodes per one point of the plane. For better illustration, consider a random network with $|\mathcal{N}| = 100$ and $M = 10$. In this case, every point contains on average a single node. Thus, the average number of nodes per point has to be 1 ($= 100/10^2$ in this example). As every node covers an area of size πr^2 , the expected average number of neighbors is calculated this way.

In contrast to the existing solutions described in Section 1, our localization concept does not rely on a distance estimation method. In other words, the node is able to receive messages from its neighbors (the set of its adjacent

nodes), but it is unable to determine their distances or positions. We also avoid a reliance on a set of predefined anchor nodes, which can be used as the localization basis. This assumption of our solution differs from many of the existing solutions described in Section 1, but it maintains the most realistic scenario. We aim to analyze the connectivity, which means no obligation of a specific hardware distance estimation capability as the node's connectivity can solely be described by the number of their neighbors. Thus, an ordinary inter node message exchange is sufficient for this purpose. On the other hand, such a simple approach has its obvious limitations. The node is hardly able to find its precise location based on its connectivity information only. As the nodes are located randomly, there is no clear relation between the node's location and its connectivity. Thus, the purpose of the algorithm's original packs-based version was to detect the nodes with a specific location within the network. In general, their connectivity is assumed to be different from the rest of the nodes. As first, we presented how to detect the nodes on the network's periphery (the border nodes) in [19]. We define them as a connected subset encircling the rest of the nodes and we assume that their connectivity tends to be weaker. Subsequently, we extended our algorithm to estimate the network's central node in [20]. We define the central node as a node that is located in a preferable position in terms of the network's connectivity ξ . This means that a significant fraction of all nodes should be located in its proximity. Thus, its connectivity is expected to be exceptionally high. An example is shown in Fig. 1. The border nodes are indicated as white circles and the central node as black. We discuss the packs-based algorithm in detail in Section 3.

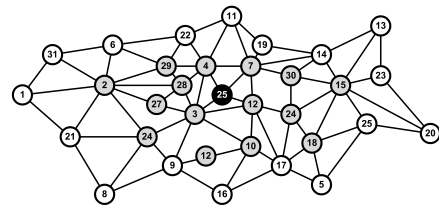


Fig. 1. A network with its border nodes (white) and a central node (black).

The following Tab. 1 list all important parameters and variables used throughout this paper to facilitate reading.

2.1 Packs

The first step of the localization algorithm is a transformation of the network, i.e., a set of the nodes $\mathcal{N} = \{n_1, n_2, \dots, n_{|\mathcal{N}|}\}$ with the cardinality $|\mathcal{N}|$ randomly distributed within a plane. In our case, the transformation describes the process of the packs' creation such that $\mathcal{N} = \{\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_i\}$. Every pack \mathcal{P}_i consists of a head (n_i in this case) and its descendants. The head is the pack's center, i.e., it is adjacent to all its descendants.

Name	Meaning
$\mathcal{N} = \{n_1, n_2, \dots, n_K\}$	network with K nodes
$ \mathcal{N} $	cardinality of network \mathcal{N}
$\mathbf{A} \in \{0, 1\}^{ \mathcal{N} \times \mathcal{N} }$	adjacency matrix $\mathbf{A}_{ij} = a_{ij}$
$d_i = \sum_j a_{ij}$	number of neighbors of node n_i
r_i	relative connectivity: (3)
ξ	network's connectivity: (2)
$\mathcal{N} = \{\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_K\}$	network with K packs
$\mathcal{P}_k = \{n_{k_1}, n_{k_2}, \dots, n_{k_K}\}$	pack with K nodes: (5)
$ \mathcal{P}_k $	pack cardinality, i.e., the number of nodes that belong to pack k
\mathcal{P}^c	central pack: (19)
c_i	bounding coordinate: (7)
c_{ij}	bounding weight between (n_i, n_j)
$R(\mathcal{P}_i)$	communication range: (6)
O_i	cohesion: (8)
\bar{O}_i	comparison basis: (9)
\mathcal{B}_N	set of border nodes
\mathcal{B}_{det}	set of detected border nodes
	Eq. (13)
\mathcal{B}_{detC}	set of correctly detected border nodes: (15)
b_i	border coordinate: (11)
b_{ij}	border weight between n_i and n_j
f	reconstruction factor: (16)
\mathcal{B}_P	set of border packs
C	corner packs: (22)
S_i	segment formed by \mathcal{P}_i : (25)

Tab. 1. List of frequently used parameters and variables.

In the heads selection algorithm presented in [19], the best connected nodes are elected as the heads by their less connected neighbors. To compare their connectivity, every node computes its *relative connectivity* as follows:

$$r_i = d_i - \frac{1}{d_i} \sum_{n_j | a_{ij}=1} d_j. \quad (3)$$

Here, d_i is the number of n_i 's neighbors. Then, n_i satisfying $r_i = \min\{r_j | a_{ij} = 1\}$ picks its best connected neighbor n_k (satisfying $r_k = \max\{r_j | a_{ij} = 1\}$) as the head and by this the pack \mathcal{P}_k is formed. We define \mathcal{P}_k 's head as follows:

$$n_k \in \mathcal{P}_k | \exists n_i | r_i = \min\{r_j | a_{ij} = 1\} \\ \wedge r_k = \max\{r_j | a_{ij} = 1\}. \quad (4)$$

Subsequently, all $n_j | a_{jk}=1$ become the descendants of \mathcal{P}_k . Then, the process is repeated for the remaining nodes (previously added nodes are not taken into account). Theoretically, a node could be adjacent to more than one head. For example, node n_i is adjacent to both n_j and n_k elected as the heads. In this case, node n_i picks \mathcal{P}_j as $r_j > r_k$. In general, a node elects pack \mathcal{P}_k with $\max r_k$ among the adjacent heads as its head. We define \mathcal{P}_k as follows:

$$\mathcal{P}_k = \{n_k, \forall n_i | r_k = \max\{r_j\}_{\mathcal{P}_j | a_{ij}=1}\}. \quad (5)$$

As a result, the better connected heads form more populated packs, as they attract also the descendants with more than

one head in its neighborhood. The process is described in even more detail in [19]. An example is shown in Fig. 2. Following the assumptions from the beginning of Section 2, we define the *communication range* $R(\mathcal{P}_i)$ of \mathcal{P}_i as follows:

$$R(\mathcal{P}_i) = \{\forall \mathcal{P}_j | \exists n_k \in \mathcal{P}_i \wedge \exists n_l \in \mathcal{P}_j; a_{kl} = 1\}. \quad (6)$$

In other words, $R(\mathcal{P}_i)$ is the set of geographically proximate packs of \mathcal{P}_i . The horizon of \mathcal{P}_i 's data acquisition is limited to $R(\mathcal{P}_i)$. Thus, all required data have to be available within $R(\mathcal{P}_i)$ for \mathcal{P}_i .

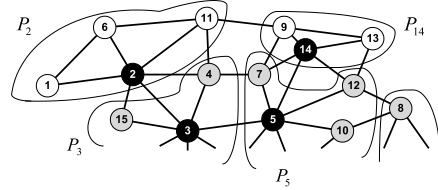


Fig. 2. A network fragmented into packs.

2.2 Network Connectivity

In [19], we defined two major factors affecting \mathcal{P}_i 's connectivity. As first, the cardinality matters. The term cardinality refers to the number of the nodes included in the pack and we use $|\mathcal{P}_i|$ to denote the cardinality of \mathcal{P}_i . The size of $|\mathcal{P}_i|$ depends on the node's density in n_i 's one hop neighborhood and on the n_i 's connectivity. Thus, $|\mathcal{P}_i|$ tends to be increasing for n_i located in a preferable position within the network. On the other hand, $|\mathcal{P}_i|$ tends to be decreasing for n_i located on the network's periphery. Another important factor is the strength of the pack's connectivity with the surrounding packs. Thus, also the number of the inter connecting nodes matters. The main idea is to emphasize the difference between $|\mathcal{P}_i|$ located in a preferable position and $|\mathcal{P}_j|$ located on the periphery as we assume, that $|\mathcal{P}_j|$ has just a limited ability to interact with $\mathcal{P}_k \in R(\mathcal{P}_j)$. Thus, we define so-called *bounding coordinates* (c_i for each node n_i in Equation (7) further ahead). The value of c_i represents the strength of n_i 's connection to $\forall n_k \notin \mathcal{P}_i | a_{ik} = 1$. In order to compute c_i , every $n_k \in \mathcal{P}_i$ assigns a weight (so-called *bounding weights*) to all its neighbors ($n_j \in \mathcal{P}_l | \mathcal{P}_l \in R(\mathcal{P}_i)$ in this case) as follows:

$$c_{kj} = \begin{cases} 1; & i \neq l \\ 0; & i = l \vee k = j. \end{cases}$$

Consequently, every $n_j \in \mathcal{P}_i$ calculates c_j as follows:

$$c_j = \frac{c_{jj} + \sum_{n_k | a_{jk}=1} c_{jk}}{d_j + 1}. \quad (7)$$

In conclusion, n_i (which is the head of \mathcal{P}_i) collects the *bounding coordinates* of its descendants and calculates the final pack's characteristics labeled as *cohesion* O_i for pack \mathcal{P}_i stated as follows:

$$O_i = |\mathcal{P}_i| \frac{c_i + \sum_{n_j | n_j \in \mathcal{P}_i} c_j}{|\mathcal{P}_i|} = c_i + \sum_{n_j | n_j \in \mathcal{P}_i} c_j. \quad (8)$$

As a result, O_i reflects both connectivity and size of \mathcal{P}_i . In order to study the pack's mutual relations, we define a so-called *comparison basis* (\bar{O}_i in the case of \mathcal{P}_i) as follows:

$$\bar{O}_i = \frac{O_i + \sum_{\mathcal{P}_j \in R(\mathcal{P}_i)} O_j}{|R(\mathcal{P}_i)| + 1}. \quad (9)$$

In other words, \mathcal{P}_i collects available data from $R(\mathcal{P}_i)$. In consequence it is able to analyze its own connectivity in the context of the surrounding packs.

3. Packs-Based Algorithm

As we explained in Section 2, our algorithm derives the location information from the node's connectivity. We use the number of the node's neighbors (notation d_i for i -th node n_i) to serve as basic connectivity information. On the other hand, such trivial information is hardly sufficient for a node's localization. In order to find a relation between the location and the connectivity, it is necessary to analyze the connectivity in a broader sense. Thus, we fragment the network into larger elements labeled as packs as defined in Section 2.1. The analysis of their connectivity and their mutual relations allows to derive objective location information. We explain these in more detail in this section.

3.1 Border Nodes Detection

A side effect of our approach is the potential to detect border nodes. We do not strive to detect all border nodes but even if some nodes discover that they are on the border of a network, they tremendously facilitate the extraction of locality information. In [19], we aimed to detect the set of the border nodes (denoted as \mathcal{B}_N) of the network. The border nodes define a connected ring for which we cannot find any of the remaining nodes geographically outside of such ring. The main idea is very simple, as we assumed that the packs on the network border tend to be weakly connected in comparison to the packs within the network. Thus, it is required to detect these weakly connected nodes. If \mathcal{P}_i satisfies the following condition, it considers itself as $\mathcal{P}_i \in \mathcal{B}_P$ (\mathcal{B}_P is a set of packs located on the network's border):

$$\bar{O}_i \leq \alpha O_i. \quad (10)$$

In this inequality, α is a so-called *minimal border pack threshold* constant. Its values are defined within $0.6 \leq \alpha \leq 1$ and we discuss its impact in Section 4. More details are presented in [19]. Subsequently, we also included two procedures that optimize the border nodes selection. Obviously, $\forall n_j \in \mathcal{P}_i | \mathcal{P}_i \in \mathcal{B}_P$ are not explicitly $n_j \in \mathcal{B}_N$. In the ideal case as shown in Fig. 1, every border node is adjacent to two other border nodes. In general, every node is most likely adjacent to a higher number of nodes. Thus, we aim to reduce the selected nodes. To do this, we use a similar approach as in (7). First, $n_i \in \mathcal{P}_i | \mathcal{P}_i \in \mathcal{B}_P$ assigns weights to its neighbors

as follows:

$$b_{ij} = \begin{cases} 1; & i = j \vee n_j \in \mathcal{P}_k | \mathcal{P}_k \in \mathcal{B}_P \\ 0; & n_j \in \mathcal{P}_k | \mathcal{P}_k \notin \mathcal{B}_P. \end{cases}$$

Here, b_{ij} is so-called *border weight*. It is worth emphasizing that only nodes $n_i \in \mathcal{P}_i | \mathcal{P}_i \in \mathcal{B}_P$ are performing these computations. Then, each n_i computes its *border coordinate* b_i as follows:

$$b_i = \frac{b_{ii} + \sum_{n_j | a_{ij}=1} b_{ij}}{d_i + 1}. \quad (11)$$

Consequently, n_i is able to recognize whether it is a suitable border node or not by examining this term:

$$\beta < |\mathcal{T}_i|; \mathcal{T}_i = \{\forall n_j | a_{ij} = 1 \wedge b_j < b_i\}. \quad (12)$$

Here, \mathcal{T}_i is the set of the n_i 's neighbors which are bound stronger to the border. If n_i has more of these neighbors than β , it does not consider itself as the border node. In order to avoid false detections, we added a second procedure:

$$\mathcal{B}_{det} = \{\forall n_i | b_i > 0.5 \wedge \exists n_j | a_{ij} = 1; b_j < 0.5\}. \quad (13)$$

It is worth mentioning that both c_i and b_i assign values from the interval $[0; 1]$. In (12) we defined an upper-bound for b_i . If n_i exceeds this upper-bound, another check is performed as defined in (13). If this condition also holds, n_i considers itself as $n_i \in \mathcal{B}_{det}$. We present the performance of the distributed algorithm in Section 4.

3.2 Central Node Estimation

An extension of the algorithm subsequently allows to estimate the position of the central node. Here, we just aim to detect the pack that is least interacting with its detected border nodes. First, we define the packs of our interest as *candidates set* \mathcal{D} . Set \mathcal{D} is stated as follows:

$$\mathcal{D} = \{\forall \mathcal{P}_i | \mathcal{P}_i \notin \mathcal{B}_P\}. \quad (14)$$

Thus, we are interested in all packs except those on the border. Then, we search for \mathcal{P}_i with the smallest b_i among $\forall \mathcal{P}_i \in \mathcal{D}$ and label it as the central node. The results of this procedure are presented in Section 4.

4. Algorithmic Behavior in Dense Networks

We tested the packs-based algorithm's performance. Here, we aim to determine its performance on a set of random, but dense networks. In other words, the network's topology is random, but we exclude the existence of significant void areas. We elected these networks as it is possible to determine the network's physical center and border explicitly, which allows the direct comparison with the true, physical location. In case of a complex, irregular topology, this is

much more complicated. In [19], we define a so-called reference algorithm, detecting the network's border as defined in Section 2. Here, the node's physical coordinates are used for the network's border guaranteeing the detection of the true border. Consequently, the subset of the correctly detected border nodes (\mathcal{B}_{detC}) is defined as follows:

$$\mathcal{B}_{detC} = \{n_i | n_i \in \mathcal{B}_{det} \wedge n_i \in \mathcal{B}_N\} \quad (15)$$

with \mathcal{B}_{det} from (13). We introduce the *reconstruction factor* f to define the $|\mathcal{B}_{detC}|$'s ratio to $|\mathcal{B}_N|$:

$$f = \frac{|\mathcal{B}_{detC}|}{|\mathcal{B}_N|}. \quad (16)$$

Before we discuss the gained results, it is important to emphasize the process of \mathcal{B}_N 's detection. The setting of α and β plays a crucial role. According to (10), α defines a threshold value. For α equals to one, the number of the detected nodes decreases in comparison to the smaller α as (10) becomes more tight. In other words, the connectivity difference between \mathcal{P}_i and $\forall \mathcal{P}_j \in R(\mathcal{P}_i)$ has to be larger for increasing α . On the other hand, it is unlikely to expect a significant difference in the connectivity among the proximate packs in a dense network. The value of β serves as an auxiliary parameter allowing us to modify the detection process as shown in (12). In fact, \mathcal{B}_{det} is derived from the subset of $\forall n_i \in \mathcal{P}_i | \mathcal{P}_i \in \mathcal{B}_p$. The value of β defines how many of them are included in \mathcal{B}_{det} as explained in (12). An example (taken from [19]) is shown in Fig. 3. In this scenario, $\alpha=1$. Here, the algorithm detects up to 55% of the true border nodes. The proportion of the detected nodes can be increased for decreasing α . For example for $\alpha=0.8$, the algorithm is able to detect up to 75% of the true border nodes. On the other hand, the false detections increase with the increase of α . We define as false detections the subset of the nodes detected by our algorithm but not by the reference algorithm ($\mathcal{B}_{det} - \mathcal{B}_{detC}$). We discussed these issues in more detail in [19]. For example, $|\mathcal{B}_{det} - \mathcal{B}_{detC}|$ is twice as large for $\alpha=1$ as for $\alpha=0.8$ for $|\mathcal{N}|=1000$.

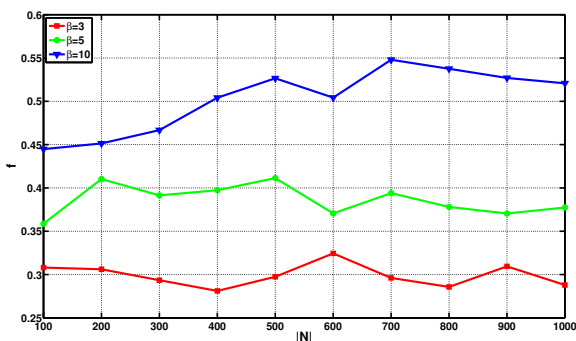


Fig. 3. The relation of the reconstruction factor f and $|\mathcal{N}|$ for $\alpha=1$.

Subsequently, we analyzed the algorithm's ability to estimate the central node. In this analysis, we are interested in the Euclidean distance of the true central point and the

detected points. We calculated the true position of a virtual center node as follows:

$$x_{p_{Real}} = \frac{\sum_{\forall n_i \in \mathcal{N}} x_i}{|\mathcal{N}|}; y_{p_{Real}} = \frac{\sum_{\forall n_i \in \mathcal{N}} y_i}{|\mathcal{N}|}. \quad (17)$$

In our simulations, every node has assigned two coordinates: x_i and y_i for n_i . These coordinates represents its location within a plane and are unique for every node. Based on them, we computed the true central point (p_{Real}) as the average coordinates from all nodes in the network. Then, we calculate the Euclidean distance between p_{Real} and the detected value (we use index p to denote the detected values) as follows:

$$E = \sqrt{(x_{p_{Real}} - x_p)^2 + (y_{p_{Real}} - y_p)^2}. \quad (18)$$

The gained results are shown in Fig. 4. Obviously, the distance among p_{Real} and p is as close as three hops even for large networks. We evaluate these results as very interesting. Without any location information, the algorithm is able to detect a significant fraction of the network's border nodes and estimate the central node with a solid precision.

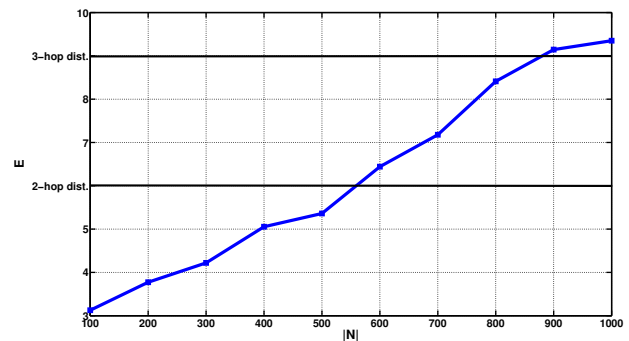


Fig. 4. The distance E from (18) as a function of $|\mathcal{N}|$.

An example for $(500, 100, 15)$, thus $|\mathcal{N}|=500$ and $\xi = 35.325$ with the selection of $\alpha=0.8$ is shown in Fig. 5, displaying a network topology in a plane showing the true physical coordinates x and y . The detected border nodes are indicated by red squares and the estimated central node by a black square. The network's border is reconstructed with acceptable precision. The major drawback is the necessity to define α . Without any topology information, this task remains difficult.

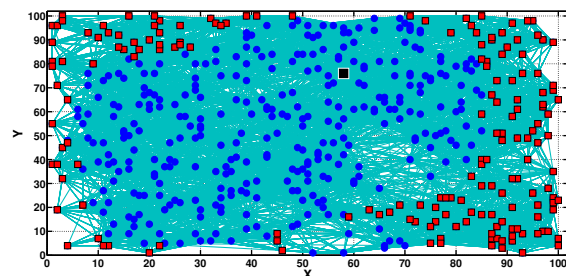


Fig. 5. The detected boarder nodes (red squares) and the central node (black square).

5. Segment-Based Algorithm

As mentioned in Section 3.2, we aim to improve our packs-based algorithm further. The packs-based version presented in previous sections is suitable for networks with only a certain degree of connectivity. Thus, this limitation is in our scope. We aim to gain acceptable performance for any network, regardless of its topology or connectivity. Moreover, the particular setting of α and β remains a limiting factor. The values mentioned in Sections 3.1 and 4 are optimized for dense networks. It is a tricky question in general how to assign their optimal values without any topology information. Thus, we avoid them at all in the segment-based algorithm. Subsequently, we also aim to extend location information provided by our algorithm. The packs-based algorithm performs a reconstruction of the network's topology as it detects specifically located nodes, such as border nodes and the central node. On the other hand, the rest of the nodes are still unaware of their location. Thus, we aim to provide all nodes with location information. In compliance with the packs-based algorithm, the methods presented in Section 2.1 and 2.2 are equally applied in the extended version, the segment-based algorithm. As opposed to the methods presented in Section 3.1 and 3.2, which are replaced completely, this makes straightforward comparisons between the packs-based and segment-based version very challenging. In the segment-based algorithm, we do not emphasize the decentralization factor. Instead of detecting a limited set of the nodes, the segment-based algorithm forms a structure in terms of packs and segments (explained further ahead) that enables to define the location of all nodes. Such a task could not be done in a completely decentralized manner. It is a much more complex task that offers significant advantages on the other hand. The most obvious advantage is its potential in routing. As the position of every node is determined, it can be used as a base for a routing decision. Due to these differences, we cannot offer many direct comparisons of their performance. In this section, we present the segment-based version, satisfying all of these new requirements.

5.1 The Detection Process

First, we present a generalized detection process. The central node location and the border nodes remain in our scope, but in a modified way, as a central pack of nodes as well as border packs. It is worth emphasizing, that we still assume the availability of connectivity information only. Thus, we aim to detect the central pack, a dominant pack in terms of connectivity, which takes over the meaning of the central node in the packs-based algorithm. As each pack has got a head node assigned, this node could take on the meaning of a central node. The relation to a geographical center is meaningless as it can be impossible to determine the geographic center of networks with an irregular topology. We simply aim to find a dominantly connected pack. The network's border can also be complicated to define. Thus, we aim to find a set of packs, located on the network's periphery

with respect to the central pack. Specifically, a set of packs with maximal distance from the central pack is in our scope. We call this set *corner packs* and label this set as C . These packs are detected instead of \mathcal{B}_P from the pack-based version. In contrast to the packs-based version, the distances are taken into account and the corner packs serve different purpose. Obviously, C is a subset of \mathcal{B}_P from the pack-based algorithm. As the result, we aim to transform the network into a form as illustrated in Fig. 6. Here, the central pack is illustrated as the black square, the corner packs as red squares and the rest of nodes as gray circles. The network is fragmented into four equivalent areas illustrated as squares. We label these fractions as segments. In this context, C serves as the set of reference points. It is also worth emphasizing that the segments are formed on top of the packs. In other words, every segment is a set of the packs. We aim to fragment the network into several areas formed around $\mathcal{P}_i | \mathcal{P}_i \in C$. We also assume that each of these segments is connected with the central pack. It is worth mentioning that we aim to form equally sized segments. Such approach promises an optimal number of segments regarding the network's size. Another important aspect is the distance to the central pack. As this pack is unique, it can serve as the reference point for a distance measurement. The dotted circles represent the hop distances from the central pack (indicated as black square) in Fig. 6. Thus, the node's hop distance defines its position within one of the circles, that is how many packs are needed to pass from one pack to another. As we mentioned in Section 5, we aim to provide every node in the network with location information. First, a node can determine its position within one of the circles. Subsequently, it can determine its position within one of the squares. We explain this process in more detail in this section.

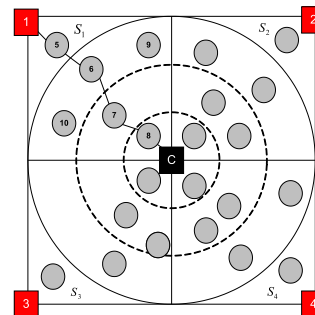


Fig. 6. Desired network structuring by segmentation: red squares denote border packs, the central pack is indicated by a black square.

The process of the network's transformation begins with the central pack's detection. We assume the invariance of the processes described in Sections 2.1 and 2.2. Thus, the network is fragmented into the pack's set ($\mathcal{N} = \{\mathcal{P}_i, \mathcal{P}_j, \dots, \mathcal{P}_k\}$). We define the pack \mathcal{P}_i as the central pack \mathcal{P}^c if \mathcal{P}_i satisfies the following condition:

$$\mathcal{P}^c = \mathcal{P}_i | \bar{O}_i = \max\{\bar{O}_j\}_{\mathcal{P}_j \in \mathcal{N}}. \tag{19}$$

Here, \bar{O}_i refers to (8). Subsequently, we define the so-called pack's distance $d(\mathcal{P}_i, \mathcal{P}_j)$ as follows:

$$d(\mathcal{P}_i, \mathcal{P}_j) = \begin{cases} 1; & \mathcal{P}_j \in R(\mathcal{P}_i) \\ m; & m = |\mathcal{W}(\mathcal{P}_i, \mathcal{P}_j)| - 1. \end{cases}$$

Here, $\mathcal{W}(\mathcal{P}_i, \mathcal{P}_j)$ is a set of packs containing all packs that are required to form the shortest path connecting \mathcal{P}_i with \mathcal{P}_j ($\mathcal{W}(\mathcal{P}_i, \mathcal{P}_j) = \{\mathcal{P}_i \dots \mathcal{P}_j\}$) and $|\mathcal{W}(\mathcal{P}_i, \mathcal{P}_j)|$ denotes its cardinality. In other words, $\mathcal{W}(\mathcal{P}_i, \mathcal{P}_j)$ is the (unordered) set of packs required to be traversed in order to connect \mathcal{P}_i with \mathcal{P}_j . For example, $\mathcal{W}(\mathcal{P}_i, \mathcal{P}_j) = \{\mathcal{P}_i, \mathcal{P}_k, \mathcal{P}_j\}$ means that $\mathcal{P}_j \notin R(\mathcal{P}_i)$, but $\mathcal{P}_k \in R(\mathcal{P}_i)$ and $\mathcal{P}_j \in R(\mathcal{P}_k)$, making $d(\mathcal{P}_i, \mathcal{P}_j) = |\mathcal{W}(\mathcal{P}_i, \mathcal{P}_j)|$ equivalent to the hop distance from pack \mathcal{P}_i to pack \mathcal{P}_j , thus in terms of packs rather than nodes. The corner packs are detected with respect to $d(\mathcal{P}_i, \mathcal{P}^c)$. In other words, every \mathcal{P}_i satisfying the following condition is considered as a corner pack:

$$d(\mathcal{P}_i, \mathcal{P}^c) = \max\{d(\mathcal{P}_j, \mathcal{P}^c)\}_{\mathcal{P}_j \in R(\mathcal{P}_i)}. \quad (20)$$

Thus, $\mathcal{P}_i \in C$ if it is not adjacent to \mathcal{P}_j with $d(\mathcal{P}_j, \mathcal{P}^c) > d(\mathcal{P}_i, \mathcal{P}^c)$. As it is adjacent to at least one other \mathcal{P}_j with the same maximal value, $\mathcal{P}_i \in C$ only if the following condition holds:

$$\mathcal{P}_k \in R(\mathcal{P}_i) \wedge \bar{O}_i < \bar{O}_k. \quad (21)$$

Thus, only the better connected packs are selected in the case, when $\exists \mathcal{P}_j \in R(\mathcal{P}_i) | d(\mathcal{P}_j, \mathcal{P}^c) = d(\mathcal{P}_i, \mathcal{P}^c) = \max\{d(\mathcal{P}_k, \mathcal{P}^c)\}_{\mathcal{P}_j \in R(\mathcal{P}_i)}$. We aim to detect a limited number of corner packs in this case. Consequently, we define C as follows:

$$C = \{\forall \mathcal{P}_i | d(\mathcal{P}_i, \mathcal{P}^c) = \max\{d(\mathcal{P}_j, \mathcal{P}^c)\}_{\mathcal{P}_j \in R(\mathcal{P}_i)} \wedge \bar{O}_i < \bar{O}_k | \exists d(\mathcal{P}_k, \mathcal{P}^c) = d(\mathcal{P}_i, \mathcal{P}^c)\}. \quad (22)$$

5.2 The Network's Segments

In this section, we explain the segment's construction. First, we define the term segment. A segment S_i is a set of packs connecting a corner pack $\mathcal{P}_i \in C$ with the central pack. Every \mathcal{P}_k included in $\mathcal{W}(\mathcal{P}_i, \mathcal{P}^c)$ becomes a member of S_i ($\mathcal{P}_k \in S_i$). Subsequently, every $\mathcal{P}_j \notin S_i$ selects its most proximate segment and becomes member of this segment. As a result, a structure similar to Fig. 6 is formed. As we consider the circles to be the packs, $\mathcal{W}(\mathcal{P}_1, \mathcal{P}^c) = \{\mathcal{P}_1, \mathcal{P}_5, \mathcal{P}_6, \mathcal{P}_7, \mathcal{P}_8, \mathcal{P}^c\}$. Also, \mathcal{P}_9 and \mathcal{P}_{10} has to be included in S_1 , as they are not part of any other path from an corner pack to the center and $\mathcal{W}(\mathcal{P}_1, \mathcal{P}^c)$ contains the closest packs included in any segment. As we aim to provide every node with location information, it is necessary to include every pack in any segment. We explain this in more detail in this subsection. Thus, the segment is formed around $n_i \in \mathcal{P}_i | \mathcal{P}_i \in C$. Moreover, we assume that every S_i has

a unique path $\mathcal{W}^U(\mathcal{P}_i, \mathcal{P}^c)$. That means that no other S_j includes any $\mathcal{P}_k \in S_i | \mathcal{P}_k \in \mathcal{W}(\mathcal{P}_i, \mathcal{P}^c)$ to its $\mathcal{W}(\mathcal{P}_j, \mathcal{P}^c)$. This property is stated as follows:

$$\mathcal{W}^U(\mathcal{P}_i, \mathcal{P}^c) = \mathcal{W}(\mathcal{P}_i, \mathcal{P}^c) | \nexists \mathcal{P}_j \in \mathcal{W}(\mathcal{P}_i, \mathcal{P}^c) \wedge \mathcal{P}_j \in \mathcal{W}(\{\mathcal{P}_k\}_{\mathcal{P}_k \in \{C \setminus \mathcal{P}_i\}}, \mathcal{P}^c). \quad (23)$$

In consequence, every S_i is connected to \mathcal{P}^c via a unique path. This also implies, that not every $\mathcal{P}_i \in C$ forms S_i . These packs ($\mathcal{P}_i \in C | \nexists \mathcal{W}^U(\mathcal{P}_i, \mathcal{P}^c)$) behave like non-corner packs and elect their most proximate segments as defined in (24). On the other hand, this guarantees an adequate number of S_i in the network with respect to the networks connectivity. In a strongly connected network, more $\mathcal{P}_i \in C$ can find $\mathcal{W}^U(\mathcal{P}_i, \mathcal{P}^c)$ than in a weakly connected network as there exist more potential candidates for $\mathcal{P}_j \in \mathcal{W}^U(\mathcal{P}_i, \mathcal{P}^c)$. Thus, this significant network's property is implicitly included without a need to be determined. The rest of \mathcal{P}_i detects their segments as follows:

$$\mathcal{P}_i \in S_j | d(\mathcal{P}_i, \{\mathcal{P}_k\}_{\mathcal{P}_k \in \mathcal{W}^U(\mathcal{P}_j, \mathcal{P}^c)}) = \min\{d(\mathcal{P}_i, \{\mathcal{P}_l\}_{\mathcal{P}_l \in \mathcal{W}^U(\{\mathcal{P}_m\}_{\mathcal{P}_m \in S_m}, \mathcal{P}^c)})\}. \quad (24)$$

Here, \mathcal{P}_i looks for the closest \mathcal{P}_k included in the path from the corner pack to the central pack. Instead of looking for \mathcal{P}_j , it is sufficient to look for $\mathcal{P}_k \in S_j$. This reduces the required communication range, especially for \mathcal{P}_i closer to \mathcal{P}^c . Consequently, we define S_i as follows:

$$S_i = \{\mathcal{P}_i \in C | \exists \mathcal{W}^U(\mathcal{P}_i, \mathcal{P}^c) \wedge \forall \mathcal{P}_j \in \mathcal{W}^U(\mathcal{P}_i, \mathcal{P}^c) \wedge \forall \mathcal{P}_l | d(\mathcal{P}_i, \{\mathcal{P}_k\}_{\mathcal{P}_k \in \mathcal{W}^U(\mathcal{P}_l, \mathcal{P}^c)}) = \min\{d(\mathcal{P}_i, \{\mathcal{P}_l\}_{\mathcal{P}_l \in \mathcal{W}^U(\{\mathcal{P}_m\}_{\mathcal{P}_m \in S_m}, \mathcal{P}^c)})\}. \quad (25)$$

We discuss the gained results in more detail in Section 6.

6. Simulation Examples

The major advantage of our improved algorithm is its general approach reaching a meaningful result regardless of the network's topology. In this section, we present the gained results based on simulations. We focus on networks consisting of $|\mathcal{N}| = 500$ nodes with a random topology very much similar to the previous example in Fig. 5. In the presented figures, we illustrate the nodes affiliated by the central pack as red circles. For the comparison with the packs-based version, we illustrate the head of the central pack as the black square, indicating the central node. The rest of the nodes obtain a color depending on the segment they are associated to. We also highlighted $\mathcal{P}_i \in C$ by red squares. In this analysis, we discuss the results gained from three different topologies.

In the first case, we present the results gained from a weakly connected network. This network is described by the triplet (500, 200, 15) resulting in a connectivity of $\xi = 8.83$. The results are shown in Fig. 7, the blue lines indicating the connections, the five colors (white, black, green, pink, yellow) the five different segments. Red colored nodes

indicate nodes affiliated by the central pack. We display the network topology in an X-Y plane showing the true physical coordinates x and y again. Obviously, there are many significant void areas in the network. The position of the central pack is not related to the network's geographic center. The network is fragmented into five segments, which cover different fractions of the network.

In the second case, we aim on a network which is much better connected than in the first case. Its ordered triplet equals to $(500, 100, 10)$ resulting in a connectivity of $\xi = 15.7$. Thus, every node n_i is expected to have in the average almost twice as many neighbors as in the first case. On the other hand, some void areas are still present. The results are shown in Fig. 8. In this case, the position of the central pack indicated by red circles is much closer to the network's theoretical geographic center. The network is again fragmented into five segments, indicated by the five colors white, black, pink, yellow and green, which cover more uniform fractions of the network than in the first case.

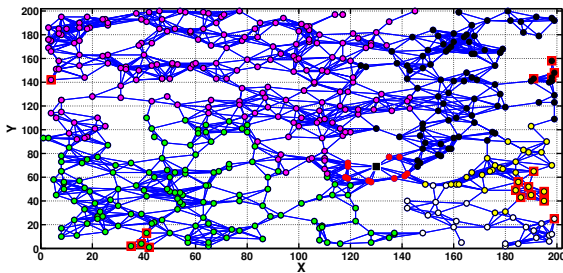


Fig. 7. A weakly connected network example. The five segments are indicated by the colored nodes in white, black, pink, green and yellow. The central pack's nodes are indicated by red nodes with the central node indicated by a black square, the detected corner packs by red squares.

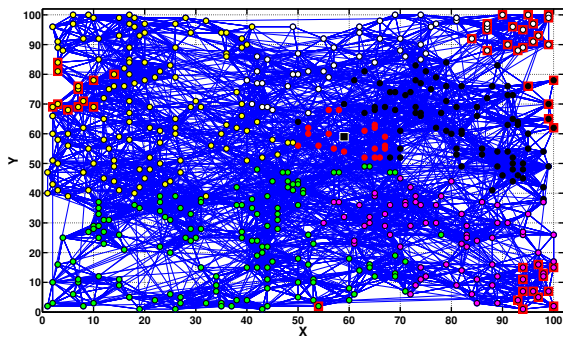


Fig. 8. A typically connected network example. The five segments are indicated by the colored nodes in white, black, pink, green and yellow. The central pack's nodes are indicated by red nodes with the central node indicated by a black square, the detected corner packs by red squares.

In the third case, we aim on a strongly connected network without any void areas. In this case, its triplet equals to $(500, 100, 15)$ resulting in the highest connectivity of $\xi = 35.325$. The results are shown in Fig. 9. Here, we applied the segment-based algorithm on the same network as in Fig. 5. The central node (black square) is again located in the proximity of geographic center and the five segments cover almost uniform fractions of the network. In contrast to the

results presented in Section 4, just a fraction of the border nodes is detected as the corner packs. This results from the definition of \mathcal{C} presented in Section 5.1. As these packs serve as the reference points for the segment construction, it is required to detect them very precisely.

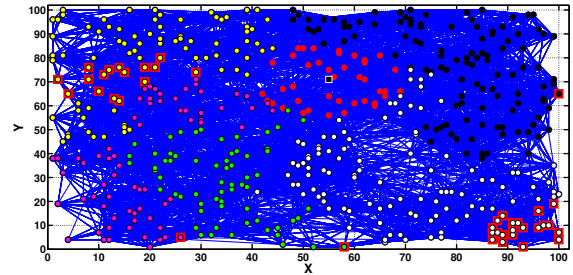


Fig. 9. A dense network example. The five segments are indicated by the colored nodes in white, black, pink, green and yellow. The central pack's nodes are indicated by red nodes with the central node indicated by a black square, the detected corner packs by red squares.

As the result, every n_i obtains location information consisting of two parts. First, every n_i is aware of the segment, in which is located ($n_i \in \mathcal{P}_i \vee \mathcal{P}_i \in \mathcal{S}_j$). Second, the distance to the central pack $d(\mathcal{P}_i, \mathcal{P}^c)$ is a metric worthy to emphasize. As \mathcal{P}^c is a unique pack within the network, these distances offers additional location data. Consider the network shown in Fig. 6. Besides of the position within one of the squares, also the position within the circles matters. This is achieved by $d(\mathcal{P}_i, \mathcal{P}^c)$. Thus, the awareness of $\mathcal{P}_i \in \mathcal{S}_j$ and $d(\mathcal{P}_i, \mathcal{P}^c)$ means very solid location data for $n_i \in \mathcal{P}_i$.

The total number of the segments in all our presented examples was five. In general, their number varies from three to six for our test networks. It is worth emphasizing that this number is related to the shape of networks we are using in simulations. Following our definitions presented in Section 5.1, the algorithm aims to detect the most distant packs. In case the network border is shaped like a perfect rectangle, the number of $\mathcal{P}_i \in \mathcal{C}$ will be four as the only packs without a more distant adjacent packs in their neighborhood are located in the rectangle's corners. Our networks are shaped as rectangles, but with an irregular border. Therefore, the number of $\mathcal{P}_i \in \mathcal{C}$ varies as the number of packs without more distant adjacent packs in their neighborhood is unpredictable in general. Also, the network's density is another important factor. In order to form a segment, every $\mathcal{P}_i \in \mathcal{C}$ has to establish its unique path to the central pack. Obviously, the number of potential unique paths is related to the network's density as their number is increasing with an increasing number of the inter node connections. Consequently, these relations to the network's topology allows the algorithm to adapt for any network without specific initial requirements. This adaptability is the most significant advantage in comparison to the algorithm's packs-based version.

Two significant upgrades are added to the improved version of our algorithm. First, every node obtains location information. This is crucial for data routing for exam-

ple, as the node's location can be efficiently used as a basis for a node's routing decision. Second, the improved version is suitable for any network, regardless its topology. In the packs-based version, the settings of α and β affect the gained results significantly. As shown in Fig. 5, the gained results are valid in the case of correct settings. The major challenge is a method to obtain these correct settings without any initial information about network's size, topology or connectivity. The improved algorithm solves these issues by avoiding any requirements for a specific settings. Regardless of the network properties, it reaches the desired performance.

7. Conclusions

In this paper we proposed a novel algorithm, tackling the localization problems in WSNs. We assumed no location data available to the nodes as we consider this as the most likely scenario. In order to avoid any specific hardware requirements, we created a so-called connectivity based approach. Here, every node requires just the number of its adjacent nodes as initial data. In terms of the hardware capabilities, the nodes have to be able to send and receive messages only. Our algorithm is build on our previous work, as we applied similar techniques as presented in [19] and [20]. Here, we proposed a fragmentation method that dissects the network into larger partitions labeled as packs. We presented also a set of parameters that allowed us to efficiently analyze their mutual relations. Based on them, we improved the algorithm further to obtain the desired performance regardless of the network topology. We also simplified the initial phase of the algorithm as the segment-based version's performance does not depend on initial settings of parameters α and β . Regarding these results, we introduced an additional layer of fragmentation on top of fragmenting into packs. The network is fragmented into so-called segments, sets consisting of packs. Every segment contains an corner pack and it is connected to the central pack. Consequently, every node obtains its location data. First, it is located within one of the segments. Second, it is aware of its distance from the central pack.

8. Acknowledgment

The research was funded by the Austrian Science Fund (FWF) in project NFN SISE S10611-N13.

References

- [1] AKYILDIZ, I. F., SU, W., SANKARASUBRAMANIAM, Y., CAYIRCI, E. Wireless sensor networks: A survey. *Computer Networks Journal*, 2002, vol. 38, no. 4, p. 393 - 422.
- [2] WATTEYNE, T., MOLINARO, A., RICHICHI, M. G., DOHLER, M. From MANET to IETF ROLL standardization: A paradigm shift in WSN routing protocols. *IEEE Communications Surveys & Tutorials*, 2011, vol. 13, no. 4, p. 688 - 707.
- [3] *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*. IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements, Part 15.4. IEEE Std., Rev. 2006, 8 Sep. 2006.
- [4] [Online] Available at: <http://www.ietf.org/html.charters/roll-charter.html>
- [5] [Online] Available at: <http://www.ietf.org/html.charters/6lowpancharter.html>
- [6] *HART Field Communication Protocol Specifications*, Revision 7.1. DDL Specifications, HART Communication Foundation Std., 2008.
- [7] STOJMENOVIC, I. *Handbook of Sensor Networks: Algorithms and Architectures*. Hoboken (New Jersey, USA): John Wiley & Sons, 2005.
- [8] PATWARI, N., ASH, J. N., KYPEROUNTAS, S., HERO, A. O. I., MOSES, R. L., CORREAL, N. S. Locating the nodes - cooperative localization in wireless sensor networks. *IEEE Signal Processing Magazine*, 2005, vol. 22, no. 4, p. 54 - 69.
- [9] DESTINO, G., SALORANTA, J., THADEU FREITAS DE ABREU, G. Sensor localization in realistic environment using ultra-wideband radio. In *Proceedings of 11th International Symposium on Wireless Personal Multimedia Communications (WPMC'08)*. Saariselka (Lapland, Finland), 2008.
- [10] BENBADIS, F., FRIEDMAN, T., DIAS DE AMORIM, M., FDIDA, S. GPSFree - free positioning system for wireless sensor networks. In *Proceedings of Second IFIP International Conference on Wireless and Optical Communications Networks (WOCN'05)*. Dubai (United Arab Emirates), 2005.
- [11] SCHURGERS, C., SRIVASTAVA, M. B. Energy efficient routing in wireless sensor networks. In *Proceedings of Military Communications Conference (MILCOM'01)*. McLean (VA, USA), 2001.
- [12] YE, F., ZHONG, G., LU, S., ZHANG, L. GRAdient broadcast: A robust data delivery protocol for large scale sensor networks. *ACM Wireless Networks*, 2005, vol. 11, no. 3, p. 285 - 298.
- [13] GNAWALI, O., FONSECA, R., JAMIESON, K., MOSS, D., LEVIS, P. Collection tree protocol. In *Proceedings of Conference on Embedded Networked Sensor Systems (SenSys'09)*. Berkeley (CA, USA), 2009.
- [14] MAO, Y., WANG, F., QIU, L., LAM, S. S., SMITH, J. M. S4: Small state and small stretch routing protocol for large wireless sensor networks. In *Proceedings of 4th USENIX Symposium on Networked System Design and Implementation (NSDI'07)*. Cambridge (MA, USA), 2007.
- [15] CARUSO, A., CHESSA, S., DE, S., URPI, A. GPS free coordinate assignment and routing in wireless sensor networks. In *Proceedings of Annual Joint Conference of the Computer and Communication Societies (INFOCOM'06)*. Barcelona (Spain), 2006.
- [16] ELHAFSI, E. H., MITTON, N., SIMPLOT-RYL, D. Cost over progress based energy efficient routing over virtual coordinates in wireless sensor networks. In *Proc. of International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'07)*. Helsinki (Finland), 2007.
- [17] WATTEYNE, T., SIMPLOT-RYL, D., AUGÉ-BLUM, I., DOHLER, M. On using virtual coordinates for routing in the context of wireless sensor networks. In *Proceedings of IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'07)*. Athens (Greece), 2007.

- [18] RAO, A., RATNASAMY, S., PAPADIMITRIOU, C., SHENKER, S., STOICA, I. Geographic routing without location information. In *Proceedings of Ninth Annual International Conference on Mobile Computing and Networking (MobiCom'03)*. San Diego (CA, USA), 2003.
- [19] KENYERES, J., KENYERES, M., RUPP, M., FARKAŠ, P. Localized algorithm for border nodes detection in WSNs. In *Wireless Telecommunications Symposium 2012 (WTS'12)*. London, (UK), 2012.
- [20] KENYERES, J., KENYERES, M., RUPP, M., FARKAŠ, P. An algorithm for central point estimation in WSNs. In *35th International Conference on Telecommunications and Signal Processing (TSP'12)*. Prague (Czech Republic), 2012.

About Authors ...

Jozef KENYERES was born in Bratislava, Slovakia in 1985. He received his Bachelor's degree and Master's degree in Telecommunications from the Slovak University of Technology in Bratislava, where he is doing his doctoral studies. Since 2009 he is working as a project assistant at the Institute of Telecommunications, Vienna University of Technology, Austria.

Martin KENYERES was born in Bratislava, Slovakia in 1988. Since 2008 he is a student of Slovak University of Technology in Bratislava. His study branch is telecommunication/signal processing.

Markus RUPP received his Dipl.-Ing. degree in 1988 at the University of Saarbruecken, Germany and his Dr.-Ing. degree in 1993 at the Technische Universitaet Darmstadt, Germany, where he worked with Eberhardt Haensler on designing new algorithms for acoustical and electrical echo compensation. From November 1993 until July 1995, he had a postdoctoral position at the University of Santa Barbara, California with Sanjit Mitra where he worked with Ali H.Sayed on a robustness description of adaptive filters with impact on neural networks and active noise control. From

October 1995 until August 2001 he was a member of Technical Staff in the Wireless Technology Research Department of Bell-Labs at Crawford Hill, NJ, where he worked on various topics related to adaptive equalization and rapid implementation for IS-136, 802.11 and UMTS, including the first MIMO prototype for UMTS. Since October 2001 he is a full professor for Digital Signal Processing in Mobile Communications at the Vienna University of Technology where he founded the Christian-Doppler Laboratory for Design Methodology of Signal Processing Algorithms in 2002 at the Institute for Communications and RF Engineering. He served as Dean from 2005-2007. He was associate editor of IEEE Transactions on Signal Processing from 2002-2005, is currently associate editor of JASP EURASIP Journal of Advances in Signal Processing and JES EURASIP Journal on Embedded Systems. He was elected AdCom member of EURASIP from 2004-2012 and served as president of EURASIP from 2009-2010. He authored and co-authored more than 400 scientific papers and patents on adaptive filtering, wireless communications, and rapid prototyping, as well as automatic design methods.

Peter FARKAŠ was born in Bratislava, Slovakia. He received Ing. degree (Diploma in Electrical Engineering) from FEI STU in 1980, CSc. degree (Ph.D. in Computer Science) from St. Petersburg State Technical University, Russia in 1987 and DrSc. degree in Telecommunication from STU in 1998. Since 1981 he has been with Dept. of Telecommunication. Over the years he has visited and worked at the Dept. of Informatics IV RWTH Aachen, Computing Center of RWTH Aachen, Dept. of Communications and Radio-Frequency Engineering Vienna University of Technology, Dept. of Mathematics, Statistics and Operational Research of the Nottingham Trent University, Dept. of Communication Systems in Lancaster University, Cooperative Research Center for Broadband Telecommunications and Networking at Dept. of Computer and Communication Engineering, Edith Cowan University, Australian Telecommunications Research Institute at Curtin University of Technology.