

The Neighborhoods Method and Virtual Polar Coordinates in Wireless Sensor Networks

A. M. Sukhov¹ & D. Yu. Chemodanov¹

¹ Samara State Aerospace University, Moskovskoe sh. 34, Samara, Russia

Correspondence: A. M. Sukhov, Samara State Aerospace University, Moskovskoe sh. 34, Samara, 443086, Russia. E-mail: amskh@yandex.ru

Received: March 5, 2013 Accepted: March 26, 2013 Online Published: March 30, 2013

doi:10.5539/nct.v2n1p19

URL: <http://dx.doi.org/10.5539/nct.v2n1p19>

This work was supported by grant RFBR 13-07-00381

Abstract

In this paper we propose the development of a neighborhoods method for finding the route with the least number of transitions. A double pass of neighborhoods allows the determination of the shortest routes, and also the solution to the local minima problem. The proposed algorithms are complemented by search algorithms for the central anchor of and rules for angle calculation, which automatically leads to the assignment of a virtual polar coordinate system. The problem of reconstruction maps for the sensors, on the basis of limited data about nodes and their immediate neighbors, is discussed. The possibility of generalizing the neighborhoods method for inter-domain routing is supposed.

Keywords: neighborhoods method in sensor networks, double pass of neighborhoods, the local minima problem, virtual polar coordinates, inter-domain routing

1. Introduction

Wireless sensor network is a distributed, self-organizing network of geographically distributed autonomous devices, using sensors for the joint control of physical or environmental conditions in different areas. Currently, wireless sensor networks are used in many areas of life, including environmental monitoring, health care, or control movement.

In wireless sensor networks technology of relayed near radio 802.15.4/ZigBee (Sokolov, 2004) is used. A ZigBee protocol stack is built on the principle of a hierarchical seven-layer model of data communication protocols, in open systems OSI (*Open System Interconnection*). The stack includes a level of standard IEEE 802.15.4, responsible for the implementation of the communication channel, program network layers, and levels of support for applications, according to a specification of the Alliance ZigBee.

It is important to understand that the 802.15.4 standard defines a range of frequencies and radio format, and ZigBee is a logical network and software stack, providing security and routing functions.

The main task of routing in wireless sensor networks is to search for optimal delivery routes from the sender to the destination node, which requires minimal resources of its member nodes. There are two main classes of routing schemes in wireless sensor networks: geographical routing, and routing based on virtual coordinates.

Geographic routing is based on the physical location information of nodes, which can be obtained with the help of a GPS (*Global Positioning System*), or by localization algorithms, such as RSSI (*Received Signal Strength Indication*), and TDOA (*Time Difference Of Arrival*) (Bachrach & Taylor, 2005). The best-known geographical routing protocol is a protocol GPSR (*Greedy Perimeter Stateless Routing*), which is used in "Greedy Forwarding" (GF) technology for packet transfer, and the appearance of alleged hole circumvention of its perimeter, according to the right-hand rule (Fang, Gao, & Guibas, 2004).

A virtual coordinate system (VCS) is a coordinate system in which each node is characterized by the coordinate vector of dimension n , which consists of the shortest number of hops to each of a set of n fixed nodes, called anchors (Caruso, Chessa, De, & Urpi, 2005; Cao & Abdelzaher, 2004; Dhanapala & Jayasumana, 2009). The number of anchors n is taken for the network dimensions. For routing in sensor networks virtual Cartesian

coordinates are usually used.

Protocols of geographic and virtual routing often use GF technology, according to which the data packet is directed to a neighboring node that is closer to the destination than the node holding the packet (Karp & Kung, 2000). GF technology allows the sending of packets over the shortest path, but its use may cause a situation known as the local minima problem (Karp & Kung, 2000). The local minima problem occurs when there is no neighbor closer to the destination than the current node holding a packet for transmission (Dhanapala & Jayasumana, 2011). In the case of holes it is assumed to bypass perimeter empty right-hand rule (Zou & Chakrabarty, 2003). In order to solve the problem of local minimum different algorithms are offered (Wuu, Li, & Kuo, 2010; Xing, Xu, Zhao, & Harfoush, 2007), including those aimed at reducing the number of hops on the basis of the neighborhoods method (Denardin, Barriquello, Campos, & do Prado, 2011). But this method has never extended to entire route.

This paper deals with the improvement of greedy forwarding (GF) technology in order, to solve the local minima problem (Mottola, Pathak, Bakshi, Prasanna, & Picco, 2007). The basis of our approach is the concept of a neighborhood (Eschenauer, & Gligor, 2002; Fekete, Kaufmann, Kröller, & Lehmann, 2005; Whitehouse, Sharp, Brewer, & Culler, 2004), which includes all neighboring nodes. Simple information, such as a list of nodes and their neighbors, will calculate the shortest route. The neighborhood method to find the route is described in Section 2 of this paper.

Section 3 deals with testing a neighborhoods method, which was created by the original software package, given an arbitrary configuration of sensor nodes and holes. This package allows the selection of a neighborhood, to find the minimum number of hops and the routes. Section 4 discusses an algorithm to find the central point for a given node configuration, and testing of the method.

Section 5 draws an analogy between the search for routing in sensor networks, and inter-domain routing in the global network. Section 6 is devoted to the implementation of a virtual polar coordinate system, to recover a sensors map in the space, while the special algorithm calculates the radius vector and the polar angle.

2. Neighborhoods Method for Route Search

Suppose that between two nodes, X and Y , a sensor network configuration can complete many routes and choose the shortest one, or in other words, the route with the fewest hops $n(X, X_1, \dots, X_i, \dots, X_{n-1}, Y)$. On this route, each subsequent node X_{i+1} is one of the nearest neighbors of considered node X_i , but it cannot connect to another previous node of route X_j , if $i < j$.

Therefore, for further search it is useful to introduce the concept of neighborhood. By a neighborhood of node we mean the set of neighbors, i.e. the sensors are located at a distance of N hops. Thus, the nearest neighbors are neighborhood level 1, and the neighbors are located at a distance of 5 hops, from neighborhood level 5 (see Figure 1).

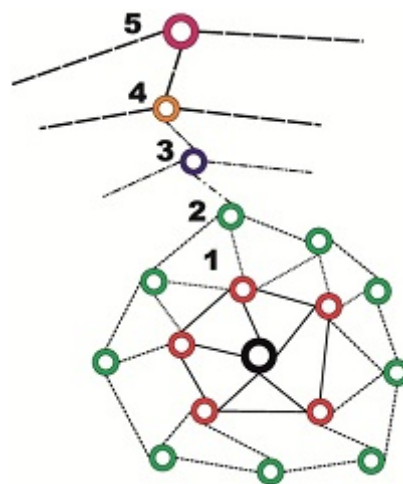


Figure 1. Illustration of neighborhoods

Initial data for the routing is a list of all possible nodes, indicating the nearest neighbors. We choose one of the two points X and Y , between which we need to install the routing, as a source (X). For this point, we will build

many neighborhoods. The neighborhood of the first level $X_1^{i_1}$ includes the nearest neighbors. In the neighborhood of the second and subsequent levels $X_n^{i_n}$ neighbors of nodes of the previous level are included, except for those that occur earlier (see Figure 2).

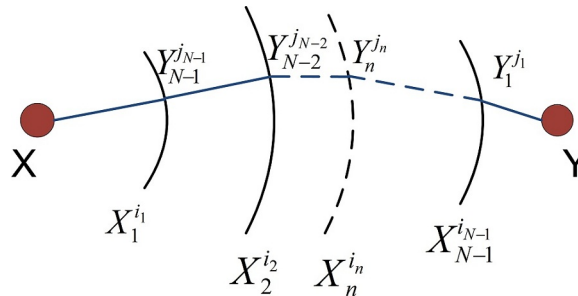


Figure 2. Illustration of the neighborhoods method

Final destination Y falls into the corresponding neighborhood $X_N^{i_N}$ in the N th step, and first phase of route search is then complete. Its result was the discovery of a set of nodes constituting N neighborhoods, and the number of hops from X to Y .

The second step is to determine the route as a list of intermediate nodes, through which data will be transferred. We must pass the search area in the reverse order.

For a node Y we construct the intersection of its first neighborhood $Y_1^{j_1}$ with $N-1$ neighborhood $X_{N-1}^{i_{N-1}}$. This set is not empty; it contains at least one node. Let us denote this node as $Y_1^X = Y_1^{j_1} \cap X_{N-1}^{i_{N-1}}$. For the node(s) we again build an intersection of its first neighborhood with the previous neighborhood $X_{N-2}^{i_{N-2}}$ of the source node X . Doing this iteration N times, we obtain a set of routes from X to Y with the number of hops N . It remains to compare the effectiveness of these routes, by matching the appropriate dynamic routing metric functions (Sukhov & Chemodanov, 2011).

In order to verify the neighborhoods method it is necessary to conduct tests on well-known and common configurations of sensors. These data can be used to find several routes using the neighborhoods method.

3. Testing the Neighborhoods Method

In order to test the proposed method of route search a special software package written in *Java* was developed. This package runs in the browser and allows the creation and testing of different configurations of sensor networks. After starting the software the menu appears on the left-hand side.

The functionality of this package allows the user to set a selected, random configuration of M sensors on the area of $K \times L$. An upper subtitle menu *Creation Area* is responsible for this function. The menu *Max X* sets K , *Max Y* is L , the number of sensors M is prescribed in the *Sensors Count*. Each sensor in a map designated circle with two boundaries, all the sensors are numbered and that number is shown in circles (see Figure 3).

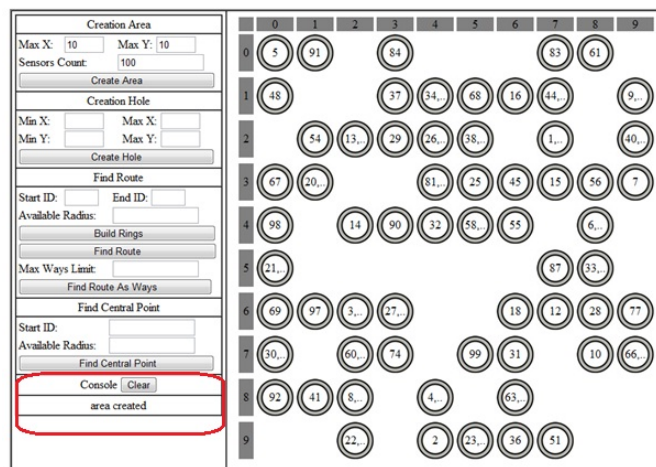


Figure 3. Provide network configuration

Additionally, section *Creation Hole* provides advanced rectangular holes for testing variants of their bypass. To do this, in fields (*Min X*, *Min Y*) and (*Max X*, *Max Y*) it is necessary to enter the coordinates of opposite vertices of a rectangle.

Find Route section is dedicated to building a route, according to an algorithm developed in the previous section of this paper. If the fields *Start ID* and *End ID* indicate numbers of the source and destination nodes of the route, the first pass (the command *Build Rings*) allows the building of the neighborhoods, and the discovery of the minimum number of transitions *N*. Figure 4 illustrates the process of breaking in the neighborhoods, indicating the number of the neighborhood in which the destination node is located (*destination sensor found in N ring*).

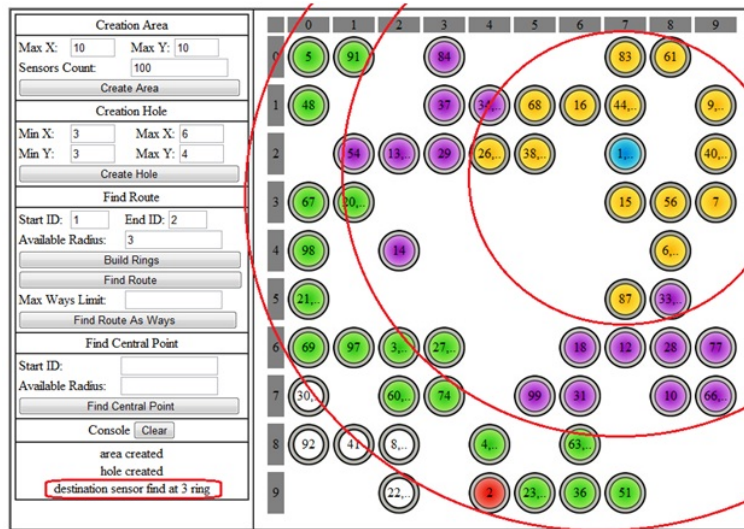


Figure 4. Partitioning of the neighborhoods

Nodes belonging to different neighborhoods are highlighted in different colors. We can also change the range of the sensors to be included in the list of the nearest neighbor (first neighborhood). This parameter is set in the *Available Radius*.

In addition, the software package can display the number of possible routes (*Max Ways Limit*), obtain a list of the sequence of nodes for alternative routes, and build these routes visually on the map (see Figure 5).

Tested configuration shows that the proposed neighborhoods method copes with the local minima problem, and allows to bypass the holes by the shortest route. For the simply connected topology of the network, when there is no separate pieces of the network, the network configuration can be completed at all times.

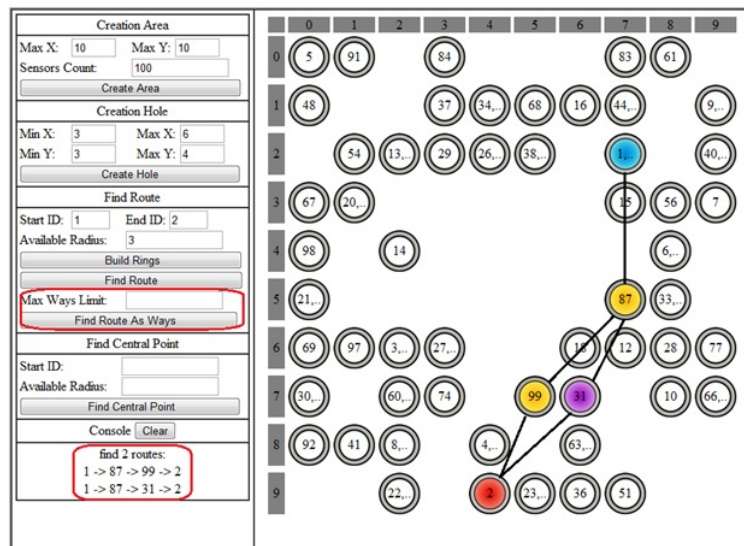


Figure 5. Find the route on the map

4. Algorithm for Finding the Center Point (the Origin)

In order to introduce virtual coordinate system (Cartesian or polar), we need to define the coordinate system origin O , a direction which will be relative to the polar angle φ , and the distance from the pole to the specified node r . Therefore, the first algorithm that should be given is a search algorithm; a central node (pole) for an arbitrary configuration of sensors.

For the central node, the number of neighborhoods covering the current configuration should be minimal. Therefore, when selecting an arbitrary node it is necessary to move it in the direction of the most remote sensors. In this case, the number of neighborhoods should decrease as long as it does not become the minimal one. Any shift from the found node should increase the number of neighborhoods.

The described algorithm can be defined as follows:

Step one: we take an arbitrary node X , build the set of neighbors that are within 1 hop, 2 hops (except for those at a distance of 1 hop), etc. So we reach the last neighborhood N .

Step two: we select any point Y from N neighborhood, construct a set of neighborhoods for this point. Let that number be L .

Step Three: we build a neighborhood of a point of X and choose the node X_1 with the minimum number of neighborhoods from $N-L/2$ neighborhood.

Step Four: Taking a new node X_1 as a pole, we construct the set of neighborhoods, the number of which is taken as the M . Any point Y of M neighborhood is chosen. After building a set of neighborhoods for this node let its number be L .

$$M=L/2 \quad (1)$$

or points X and X_1 belong to the same neighborhood, relative to the node Y , then the node on the first step of the iteration can be considered central. If this condition is not satisfied, then we denote the set M of N and then go to step 2, etc.

Algorithm will be executed as long as there is no equality (1).

Algorithm in the form of a block diagram is shown in Figure 6.

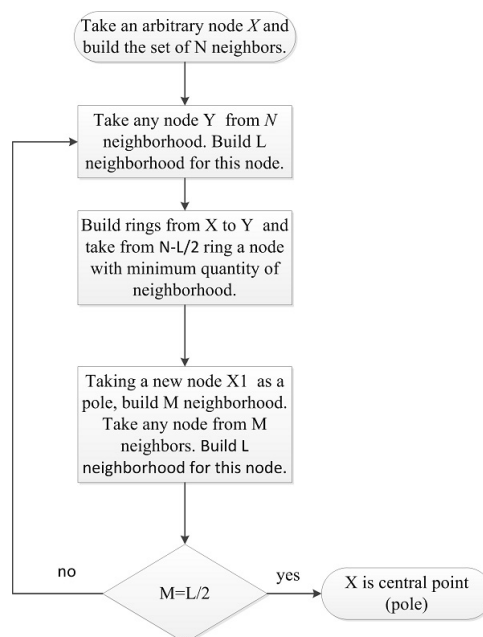


Figure 6. Algorithm for finding the center point

Figure 7 illustrates the algorithm on an example of an arbitrary sensor configuration.

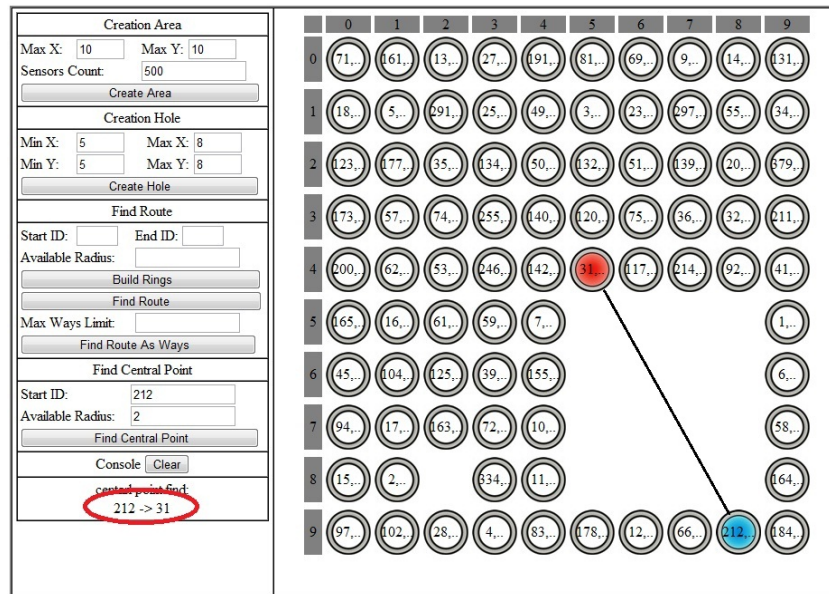


Figure 7. Finding the center point

5. Metric of Inter-Domain Routing

For experimental verification of our assumptions it should be noted that the method of neighborhoods can not only be used for sensor networks. The described method of finding the route can also be used in the process of inter-domain routing for autonomous systems (AS). It is well known that the global routing table (BGP) contains a list of autonomous systems (Rekhter & Li, 1995) and their immediate neighbors, which can take the free passage of traffic.

However, the BGP table of AS must be modernized for the application of our method. Except for the list of the nearest AS, it is necessary to specify the number of hops for each system. Let us denote Z_l as number of investigated AS, then Z_m are numbers of the neighboring AS, and k_m is the number of hops to this system. Consequently, the global routing table can be represented as $(Z_l, Z_m, k_m, \text{where } m = \overline{(1, L)})$, and L is the number of neighboring autonomous systems of Z_l .

When splitting a set of AS in the neighborhood, the number of the new neighborhood is obtained by summing the number of the current neighborhood and the number of hops k_m . If the AS Z_l belongs to an n neighborhood $X_n^{l_n}$, relative to the source of a route, then the autonomous system Z_m will be included in the $n+k_m$ neighborhood. Thus the number of neighborhoods will be much greater than the number of transit autonomous systems. The preferable route has fewer hops from the source to the destination.

Taking advantage of the global routing table we can find the best route between two or three fixed stand-alone ASs to multiple external IP addresses, using our method. We can then compare these routes with the route obtained by standard inter-domain routing BGP4.

6. Polar Coordinate System

Polar coordinates except routing problem have another important mission. This is a visual representation (map) of nodes located in the space (Dhanapala & Jayasumana, 2011). Initial information about the network configuration is extremely sparse, and contains only a list of all network sensors with their nearest neighbors (Nagpal, Shrobe, & Bachrach, 2003; Newsome & Song, 2003). On the basis of this information, can we build a complete map of the network, close to the real map?

The standard method used in sensor networks is a method of virtual coordinates. For routing in sensor networks the researchers usually use virtual Cartesian coordinates, but the neighborhoods method is suited, ideally, to polar coordinates.

In polar coordinates, each node is characterized by the radius vector r , the length of which is measured by the number of hops from the origin O and angle φ . The central point is chosen as the origin, and its location algorithm is presented in Section 4. The angle is conducted from a certain direction, taken as the direction of reference. More information on how to set the angle is presented below.

In order to set the initial direction, we choose a point X , located at a distance of m hops from the origin O . For any point Z it is easy to find the number of hops n and l of the points O and X under construction of a set of neighborhoods (see Figure 8). Denoted by $\varphi(\overline{OX}, \overline{OZ})$ polar angle, what is easy to find from the cosine theorem:

$$\varphi = \arccos \frac{m^2 + n^2 - l^2}{2mn} \quad (2)$$

where $m = |OX|$, $n = |OZ|$, $l = |XZ|$.

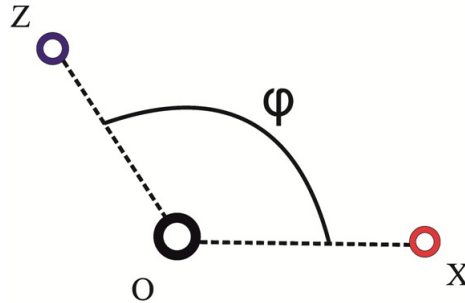


Figure 8. Polar coordinate system

However, the cosine theorem is not possible to determine the angles on the whole plane unambiguously, as the clockwise and counterclockwise angle will not vary. For an unambiguous determination of the polar angle, we should select an additional point of Y , for which $\varphi(Y) = \pi/2$, or $\cos \varphi = 0$.

The angle $\psi(\overline{OY}, \overline{OZ}) = \varphi - \pi/2$, measured from axis OY , uniquely determines the value φ of the entire domain $0 \leq \varphi < 2\pi$:

$$\varphi = \arcsin \frac{k^2 + n^2 - g^2}{2kn} \quad (3)$$

where $k = |OY|$, $g = |YZ|$. Equations (2) and (3) provide two different values for the angle φ . In constructing the sensors maps, the average of the two values should be used. Signs of argument of the functions (2) and (3) determine a quarter of the required angle φ .

Thus, for any point $Z(n, \varphi)$ polar coordinates are defined, that allow the building of a sensors map, which should reflect the existing configuration. Further research is needed to compare the resulting configuration with the actual map, to identify ways of improving the accuracy of predictions for the average deviation of the coordinates:

$$\Delta Z = \frac{1}{N} \sum_{i=1}^N |Z_i^r - Z_i^{vpc}| \quad (4)$$

where Z_i^r are the real coordinates of nodes, and Z_i^{vpc} are the virtual coordinates, and N is number of sensors in the configuration. In order to improve the accuracy of a virtual polar coordinate system, it is necessary to offer an alternative way of measuring the length of the radius vector n . The described method can be generalized easily to the 3D arrangement of nodes.

7. Conclusions and Future Studies

This work develops a neighborhoods method, which was applied previously to routing problems in wireless sensor networks. A new method is developed and tested that allows the determination of the shortest route by building neighborhoods of the original node. It is shown that the local minima problem can be resolved successfully, and we do not need to schedule an additional traversal algorithm of holes.

In order to restore the map of the sensor location, according to the nearest neighbors of the site, additional experiments are required. They will help to find an expression for the accuracy of the node position, as well as ways to improve this accuracy. All results describe the 2D configuration of sensors, but the basic algorithms and formulas can be easily generalized for the case of 3D.

A separate experiment is required for consideration of inter-domain routing. The first stage the neighborhoods method can be used to analyze existing BGP tables, to compare the routes obtained by the neighborhoods

method with those found by the standard routing. We also need to offer the best data format, to determine the neighborhood of an AS, to include data on the hops number. We can also find the center point of the current configuration of AS, and build a map of its location.

References

- Bachrach, J., & Taylor, C. (2005). Localization in sensor networks. *Handbook of sensor networks: Algorithms and Architectures, 1*. <http://dx.doi.org/10.1002/047174414X.ch9>
- Cao, Q., & Abdelzaher, T. (2004). A scalable logical coordinates framework for routing in wireless sensor networks. In *Real-Time Systems Symposium, 2004. Proceedings. 25th IEEE International* (pp. 349-358). IEEE. <http://doi.acm.org/10.1145/1218556.1218561>
- Caruso, A., Chessa, S., De, S., & Urpi, A. (2005). GPS free coordinate assignment and routing in wireless sensor networks. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE* (Vol. 1, pp. 150-160). IEEE. <http://dx.doi.org/10.1504/IJWMC.2006.012472>
- Denardin, G. W., Barriquello, C. H., Campos, A., & do Prado, R. N. (2011). A geographic routing hybrid approach for void resolution in wireless sensor networks. *Journal of Systems and Software, 84*(10), 1577-1590. <http://dx.doi.org/10.1016/j.jss.2011.03.030>
- Dhanapala, D. C., & Jayasumana, A. P. (2009). CSR: Convex Subspace Routing protocol for wireless sensor networks. In *Local Computer Networks, 2009. LCN 2009. IEEE 34th Conference on* (pp. 101-108). <http://dx.doi.org/10.1016/j.comcom.2011.01.005>
- Dhanapala, D. C., & Jayasumana, A. P. (2011a). Anchor selection and Topology Preserving Maps in WSNs—A Directional Virtual Coordinate based approach. In *Local Computer Networks (LCN), 2011 IEEE 36th Conference on* (pp. 571-579). <http://dx.doi.org/10.1109/LCN.2011.6115519>
- Dhanapala, D. C., & Jayasumana, A. P. (2011b). Geo-logical routing in wireless sensor networks. In *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2011 8th Annual IEEE Communications Society Conference on* (pp. 305-313). <http://dx.doi.org/10.1109/SAHCN.2011.5984912>
- Eschenauer, L., & Gligor, V. D. (2002). A key-management scheme for distributed sensor networks. In *Proceedings of the 9th ACM conference on Computer and communications security* (pp. 41-47). ACM.
- Fang, Q., Gao, J., & Guibas, L. J. (2004). Locating and bypassing routing holes in sensor networks. In *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies* (Vol. 4, pp. 2458-2468). <http://dx.doi.org/10.1007/s11036-006-4471-y>
- Fekete, S. P., Kaufmann, M., Kröller, A., & Lehmann, K. (2005). A new approach for boundary recognition in geometric sensor networks. *arXiv preprint cs/0508006*. http://dx.doi.org/10.1007/978-3-642-20662-7_33
- Karp, B., & Kung, H. T. (2000). GPSR: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking* (pp. 243-254). ACM. <http://dx.doi.org/10.1145/345910.345953>
- Mottola, L., Pathak, A., Bakshi, A., Prasanna, V. K., & Picco, G. P. (2007). Enabling scope-based interactions in sensor network macroprogramming. In *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE International Conference on* (pp. 1-9).
- Nagpal, R., Shrobe, H., & Bachrach, J. (2003). Organizing a global coordinate system from local information on an ad hoc sensor network. In *Information Processing in Sensor Networks* (pp. 553-553). Springer Berlin/Heidelberg. http://dx.doi.org/10.1007/3-540-36978-3_22
- Newsome, J., & Song, D. (2003). GEM: Graph EMbedding for routing and data-centric storage in sensor networks without geographic information. In *Proceedings of the 1st international conference on Embedded networked sensor systems* (pp. 76-88). ACM. <http://dx.doi.org/10.1145/958491.958501>
- Rekhter, Y., & Li, T. (1995). *A border gateway protocol 4 (BGP-4)*.
- Sokolov, M. A., (2004). Software and hardware-based wireless networks technology ZigBee/802.15.4. *Elektronnye komponenty, 12*, 80-87 (in Russian).
- Sukhov, A. M., & Chemodanov, D. Y. (2011). Variation principle and the universal metric of dynamic routing. *arXiv preprint arXiv:1112.3535*.
- Whitehouse, K., Sharp, C., Brewer, E., & Culler, D. (2004). Hood: a neighborhood abstraction for sensor networks. In *Proceedings of the 2nd international conference on Mobile systems, applications, and services*

- (pp. 99-110). ACM. <http://dx.doi.org/10.1145/990064.990079>
- Wuu, L. C., Li, W. B., & Kuo, W. C. (2010). Detour Routing Protocol for Geographic Sensor Networks. In *Broadband, Wireless Computing, Communication and Applications (BWCCA), 2010 International Conference on* (pp. 505-510). <http://dx.doi.org/10.1109/BWCCA.2010.122>
- Xing, F., Xu, Y., Zhao, M., & Harfoush, K. (2007). HAGR: Hole Aware Greedy Routing for Geometric Ad Hoc Networks. In *Military Communications Conference, 2007. MILCOM 2007. IEEE* (pp. 1-7). <http://dx.doi.org/10.1109/MILCOM.2007.4455206s>
- Zou, Y., & Chakrabarty, K. (2003). Sensor deployment and target localization based on virtual forces. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*(pp. 1293-1303). <http://dx.doi.org/10.1145/972627.972631>