DE GRUYTER OPEN
Information Technology and Management Science doi: 10.1515/itms-2015-0003
2015 / 18

Method for Balancing Energy through the Mobility of Node Agent in Mobile Sensor Network

Aleksejs Jurenoks[1], Leonids Novickis[2]
[1, 2] *Riga Technical University*

Abstract **– In the recent years, low power computing systems have gained popularity. Networks, which use low power computer systems and transmit data by using wireless connection, are called wireless sensor networks. Nowadays, the most topical studies are aimed at grouping wireless sensor networks by the new optimisation of structure of network transmission protocol, the routing optimisation in a transmission network, optimisation of network structure, as a result of which it is possible to increase the life cycle of wireless network sensors.**

There are a number of methods that allow solving this problem. These include the choice of the capacity of individual battery, the deployment of the node density, the adjustment of power transmitter, the application of energy-efficient data transfer protocol, positioning of network nodes and other methods that are associated with the introduction of additional network costs.

The present article discusses a new method for balancing energy through the mobility of network node intellectual agent, which provides the opportunity for reconfiguration of dynamic network or change of network topology.

Keywords **– Data processing, life cycle, network agent, sensor network.**

I. Introduction

Sensor networks are primarily intended for the collection of information [6]. This means that in any network there are terminal nodes for collection of information, the router nodes for transmission of information, and one or more coordinator nodes, to which all the collected information across the network is sent. The nodes pertaining to information storage and processing are mainly equipped with an uninterrupted power supply, independent distribution network connectivity and high performance computing system. This means that the direction of the data (Fig. 1) is definitely known in the sensor network, the information from the data collectors (terminals) is sent to the information storage and processing nodes by using existing routers of the network. The capacity of data traffic near a coordinator node is much higher than in the distant points [13].

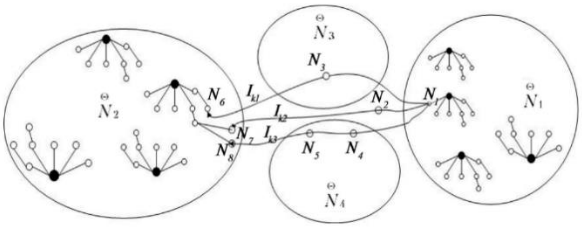

Fig. 1. The default structure of the sensor network.

The differentiated energy consumption in hubs [12], [17] always appears in an operating sensor network; as a result, the existing elements close to processing nodes faster than others stop operating due to lack of electricity and, therefore, the network ceases its overall work.

II. Related Works

To align all the consumed energy in the network nodes, a variety of energy balancing techniques are offered in many studies [8], [14], [16].

The architecture of the heterogeneous network requires different methods to improve energy efficiency.

1. The capacity of the battery in the adaptation is based on the location of the element in the network [14]. In this case, the nodes, which transmit more information, are equipped with a more powerful battery to ensure smooth power consumption across the network. This approach is easy to implement in life, but it has significant drawbacks – low network scalability and poor adaptation of functioning change. Moreover, the technical specification and design solutions of the various nodes significantly raise the cost of the network.
2. The differentiated density of network node is dependent on the estimated transmission density in a specific area [2]. The aim of this solution is to create redundancy in the network structure and provide duplication of a separate node. As a result, in the case of denial of one of the routers, its functions will be separated to another node duplication which until that moment was not used and was in sleep mode with small power consumption.

At the level of software, routing protocols are mainly used; they are based on the remaining power in the nodes [3], the use of virtual coordinates [6], distant and close turn of transmission [9], node positioning [7] as well as the use of clusters [11].

It is known that in the protocols of traditional network router metrics are used to increase network transmission capacity and prevent data transmission delays [3], [8]. The functions of the metrics may perform various intermediate network nodes to the final aim, the communication channel capacity, and the line load level [10]. In the sensor networks, node remaining energy metrics are often used to track the data processing unit. In this case, from different alternative routes one will be selected, on whose nodes there is a greater amount of the remaining energy.

©2015 Aleksejs Jurenoks and Leonids Novickis. This is an open access article licensed under the Creative Commons Attribution-NonCommercial-NoDerivs License (http://creativecommons.org/licenses/by/4.0), in the manner agreed with De Gruyter Open.

The perspective balancing method is considered to be the use of the mobility of the network components. In various studies [8], [16], the authors showed that the use of mobility brought more benefits to the increase of autonomous operation of the sensor network. Therefore, this approach will be thoroughly discussed in the following paper.

III. Planning of the Movement Pertaining to Dynamic Coordinator Node

One of the factors that impacts the change of the network configuration is change of the position of the coordinator in the network [5], [8]. The implementation of the mechanism of the network topology change will increase the amount of energy remaining in the network nodes. Operating network or network segment, there is one coordinator, to which the data flow is sent. As a result, the entire network operates according to the directed graph structures [4].

When planning the movement of the coordinator, the most important factor to be taken into account is the reduction of the amount of network reconfiguration [8].

A. The Mobility of the Network Coordinator

In the operating system, there are two possible coordinator changes:

- The expected location of the coordinator – the coordinator operates cyclically, the path of the coordinator and the time spent in each position are determined in the network;
- Random Coordinator Location – the choice of the location pertaining to the coordinator is carried out by each user while contributing to the start-up cycle. The time that is spent by coordinator in the same position depends on the user's requirements.

Using the estimated location of the coordinator it is possible to determine at which point the coordinator will be at each stage of operation.

Suppose that the system operates using cyclic movement of the coordinator through the defined trajectory, the duration that is necessary for the coordinator to pass all the trajectory is:

$$t_{net} = \{t_{net}^1, t_{net}^2, \ldots, t_{net}^m\}, \quad (1)$$

where: t_{net}^m is time during which the coordinator is in the concrete position.

The location of each coordinator is associated with the new network configuration. The system identifies the absence of coordinator in the path structure and performs reconfiguration of the network by providing a new path from the nodes to the coordinator.

Let us look at the possibility of the control of network mobility by carrying out the reconfiguration of the network. Suppose that the original location of the coordinator is known and the path is strictly defined. If the coordinator moves in the network, the position of finding a space $V_{(x,y)}^m$ changes. On the basis of (2) pertaining to energy consumption of the network node, it is known that the object transmitting data consumes power, which depends on the distance, not less than the indicated technically P_{tx} and P_{rx}.

$$P_{tx} = \begin{cases} P_{tx}\,, & K_r = 1 \\ P_{tx} * \frac{(\frac{4\pi\sqrt{P}}{\lambda * r^2})^2 * \lambda^2 * K_r{}^4}{(4\pi)^2}, & K_r > 1 \end{cases} \quad (2)$$

where: P_{tx}– the average consumed energy in the transmission mode (W); λ – the wavelength; K_r – the coefficient of working range against a standard working range, P – the power coefficient of transmitter.

When planning the structure of the network and the location of the node, as well as knowing the trajectory of coordinator movement, there is an opportunity to split the network structure into segments S_{net} . The result may be adopted:

$$S_{net} \subseteq \{V_{(x,y)}^1, V_{(x,y)}^2, \ldots, V_{(x,y)}^m\}. \quad (3)$$

The existing power of the elements of the transmitter pertaining to one of the peaks/vertices of the segment S_{net} situated in the coordinator will not change, as the operating range is not changed.

Sensor network can be defined as mathematically oriented graph G, where each graph G consists of peaks $V = \{1,2, \ldots . m\}$, edges $U: U \subseteq V * V$ and the total energy consumed in all nodes of P.

$$G = \{V, U, P\}. \quad (4)$$

V_s indicates the potential graph in the position, in which the network can be located. The condition/situation of the graph in our case is determined by the location of the coordinator in the network segment.

The consumed power by graph G in each operating cycle t_c depends on the active sector S_{net} . It is not appropriate to make reconfiguration of the network in each operating cycle of the network if the segment remains unchanged. If the agent does not make the reconfiguration of the network due to the remaining power E_n, the reconfiguration of the network will take place if the segment, in which the co-ordinator is located, S_{net} changes.

$$S_{net}^n \neq S_{net}^{n-1}. \quad (5)$$

In case of the operation of the second scheme, the coordinator can use randomly selected network location, which can be determined by the user.

$$V_{(x,y)}^a = \overline{V_{(x,y)}^1 .. V_{(x,y)}^n}. \quad (6)$$

In the results, the additional condition that contributes to network reconfiguration appears:

Network topology is changed in case the condition of the accomplishment (the position/location of the coordinator changes):

$$V_{(x,y)}^a \neq V_{(x,y)}^{a-1}. \quad (7)$$

IV. The Solution Method of the Dynamic Mobility Coordinator

A network agent, which controls the conditions that carry out the reconfiguration of the sensor network in the nodes, has been introduced into the operating system [8]:

1. The place of location pertaining to coordinator changes – $V^a_{(x,y)} = \overline{V^1_{(x,y)} .. V^n_{(x,y)}}$.
2. The large difference of the remaining energy in the network elements is found – $p_{min} > \overline{\sum_{k=1}^{n} P_k}$.

While the system operates, the agent is entrusted with two tasks:

- to ensure the collection of remaining amount of energy in the nodes and storage at the specified step t^n_c.
- to determine the period t^n_c during which it is necessary to carry out the reorganisation of the routing path.

The mathematical agent, in contrast to the network sensor elements, is multifunctional. Some tasks of the agent can be executed simultaneously. Each executable cycle of the agent or condition is realised as the independent operation. Within the agent, there is a cyclically operating activity scheduler that regardless of the situation carries out the determined tasks of the agent within the network. This approach allows coordinating the performance of the agent and managing/controlling every executable process.

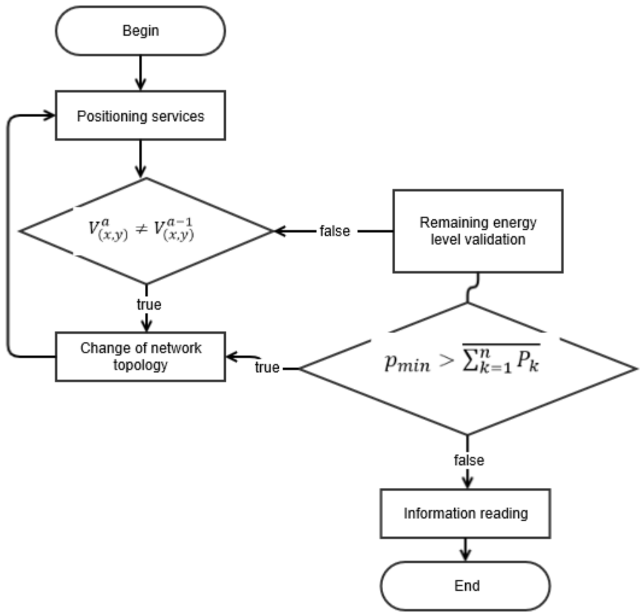

Fig. 2. The algorithm of the performance pertaining to the network agent.

In the proposed model, for each network node there is an environment pertaining to the local operating coordinator agents, where the software agents interact with one another (Fig. 2).

A. The Localisation Mechanism of the Network Agent

The monitoring and change of the network status are carried out by the network agent that operates at the protocol level. The localisation mechanism of the agent is split into four phases:

- monitoring phase;
- agent distribution phase;
- communication phase;
- operating phase.

Agent Monitoring Phase

Agent performance management module (APM) in real time, using the technical information of the network protocol, assesses the intensity of the agent communication. APM module can enumerate and control the received and sent amount of information to each separate agent in the network. With a user-specified intensity the agent performance management module determines the dependence of communication between the agent $\underline{i}$ and N agents of the node during the period $t \in T = [t_1, t_2]$ using the following formula:

$$C_{iN}(t) = a\left(\frac{M_{iN}(t)}{\sum_k M_{ik}(t)}\right) + (1-a) * M_{iN}(t-1) \tag{8}$$

where $M_{iN}(t)$ – the amount of information sent from the agent i to the node N in the time interval $T = [t_1, t_2]$,

a – the coefficient, which highlights the importance of new information regarding the old information.

Agent Distribution Phase

The agent carrying out the agreed (as defined above) monitoring activities determines the value of communication dependency coefficient between agent *i* and all the network nodes. Communication dependency coefficient R_{iN} between the agent *i* and the node *N* is determined with the following formula:

$$R_{iN} = \frac{C_{iN}}{C_{im}}, where\ N \neq m. \tag{9}$$

When the maximum value R_{iN} exceeds the defined limit coefficient θ, the distribution phase (APM) of the agent includes the deployment of the examined agent of the remote system node in the agent group:

$$k = arg_j \max(R_{iN}) \wedge (R_{ik} > \theta) \rightarrow a_i \in G_k, \tag{10}$$

where:

a_i is the copy of the agent,

G_k – agent's k group,

arg_j – functions, which return value j equal to the maximum communication value of the coefficient R_{iN}.

Agent Communication Phase

Agents operating in the network communicate by using the technical information of OSI protocol level. The agent's ability to communicate and transmit the required network information is analysed by using the obtained information from (11). The common agent's communication model can be defined as:

$$Mi = \{Sa, Se, Sp, L, Sra\}, \tag{11}$$

where: Sa – the number of agents $Sa = \{a_1, a_2, .. a_n\}$ that participate in the communication;

Se – the number of communication environment sectors where the operation takes place;

Sp – the number of network congested sectors, where nodes consume the largest amount of energy;

L – the length of the path from the coordinating point till the agent in the network;

Sra – the number of routing nodes in the path, which the agent will use for the transmission of information to the management node.

Agent Operating Phase

The task of the agent is to initialise the change of the network topology by taking into consideration two requirements:

- total network energy consumption exceeds the limit $p_{min} \geq \overline{\sum_{k=1}^{n} P_k}$;
- agent'sR_{iN} value exceeds the defined limit ratio θ.

V. Methods for Route Making

It is impossible for the systems working in reality to get correctly all the information needed in order to accurately determine the total length of the network life cycle. Some values pertaining to the network model base (e.g., capacity of transponder) may vary depending on the time. Let us look at some examples that can cause similar problems:

- In some network segments, there are technical problems that reduce the quality of transmission. This may be due to the activity of another network using the same frequency. In this case, there is the increasing necessity for retransmission of information; as a result, the consumption of energy in the problematic area increases substantially.
- External climatic factors on the autonomous power supply that, as a result, can increase power consumption in the standby mode.

As it has already been mentioned [8], [16], a wireless sensor network can be depicted by graph *G*. On the basis of the graph structure (Fig. 3), each of the graph paths (*v*, *u*) may be used for transmission of information from node *v* to node *u*.

For each of V nodes there is the defined number of adjacent objects $(v) \subset V$ that can communicate with v node. Generating the path of the network, the retransmission of broadband [15] is used; the information is simultaneously transmitted from node $N(v)$ to the entire network.

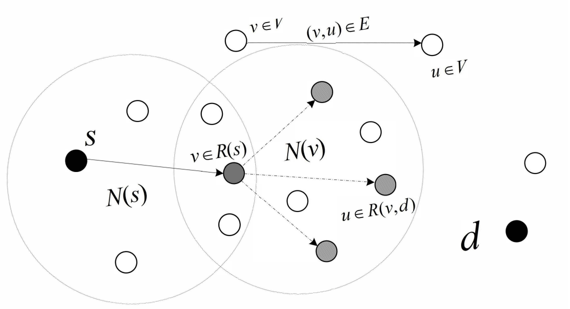

Fig. 3. Transmission of information to coordinator *d*.

The following flow algorithm in the network is proposed. Node V, receiving the information transmitted from node S, which is designed for the receiving node D, uses the following sequence of actions:

1. From $N(v)$ clusters the potential transponders are selected and sorted together according to priority$R(v,d) \subset N_v$.
2. The potential repeater group belongs to node group $R(v,d) \in N_v$ and determines the operating strategy of node *v*.
3. The information about the functionality of the node is recorded in the service of the transmission packet field, and distributed to the network management agent.

In determining priorities for retransmission, the remaining amount of energy in each individual node is taken into account. The agent selects the transponder with the greatest amount of energy remaining in node E_0. The path-building algorithm finishes its operations once it has reached the recipient node *d*.

It has been mentioned in the article that each element of the network can operate in two modes (terminal or router). To divide the tasks in the network at the initial stage of reconfiguration stage, each node is in the covering mode. During the active time the coefficients of the node are evaluated (E_0 – the amount of energy at the start of reconfiguration).

The path of wireless sensor network from the terminal s to the coordinator $d - S(s,d)$ is a deterministic structure that is built by using a recursive approach from the terminal to the coordinator. At the first stage, the network possible router set $R(s,d) \in N(s)$ is defined. This will allow describing the path $S(s,d)$ as the sorted list of lower path:

$$S(s,d) = [S(v_1,d), \ldots, S(v_n,d)], v_j \in R(s,d) \qquad (12)$$

The next step is to set up the local operation strategy for each router $R(v_j,d)$. As a result, each list pertaining to the lower path can be improved:

$$S(v_j,d) = [S(v_{j,1},d), \ldots, S(v_j,m,d)], v_{j,i} \in R(v_j,d) \qquad (13)$$

The path from the terminal up to the coordinator can be displayed as the second level included in a list of tree:

$$S(s,d) = \left[[S(v_{1,1},d), \ldots, S(v_{1,m1},d)], \ldots, [S(v_{n,1},d), \ldots, S(v_{n,m_n},d)]\right] (14)$$

When creating the route, its level is extended as long as in the list of the router the only coordinator node *d* remains.

In our case, the sensor network segment is equipped with a single coordinator node. Henceforth, let us assume that the recipient node *d* (coordinator) is fixed and will not use it in formulas.

Let us introduce the way to combine the procedure. Suppose that $S(v)$ – a path from node v to coordinator router – is in the list $R(v) = (v_1, v_2, \ldots, v_n)$ where node u does not exist. Suppose that *S(u)* is a path consisting of the excluded from the list *R (v)* nodes. Let us define the path $S(v) \in S(u)$ that uses the combined list, where nodes u are used with a low-level coefficient $-R(v) = (v_1, v_2, \dots, v_n, u)$.

A. Use of Metrics for Path Determination

Suppose that for the number of paths to node *d* there are metrics $l(S(s))$, the value which is in the indexed amount and can be compared. In similar models, the metrics are built using a recursive approach from top to bottom in defining the establishing conditions in the route [6].

$$l(S(v) = F(l(S(v_1), l(S(v_2), \dots, l(S(v_n)),\ \ v_j \in R(v). \quad (15)$$

For a wireless sensor network routing algorithm to be optimal, it is necessary to meet three conditions: The importance of the router determination. Node u is as useful as node v, if the condition is fulfilled:

$$l(S(v) \oplus S(u)) < l(S(v)). \quad (16)$$

This condition fulfils the important role. From the list of metrics S (v), all the metrics with low indicators will be selected. Consequently, the possibility to entrust the role of the router to the node with a small amount of power left will be excluded.

Preservation of priorities. Metrics remain a priority if from $(S(u)) \le l(S(v))$ resulting $l(S(u) \le l(S(v) \oplus S(u))$ to all $\forall v \in V$ and $\forall u \in N(v)$. This means that before adding router u to the list S (v) it will be evaluated whether it will be better than any other from the list S (v).

The determination of the optimality for the routers. It determines the behaviour of metrics through the full reconfiguration of the list pertaining to the router. Suppose that $R(v) = (v_1, v_2, \dots, v_n)$ – the number of routers for node v. Let us introduce an additional list with nodes $R^*(v)$ that are arranged in ascending order of metrics – $l(S(v_n)) \le l(S(v_{n+1}))$. The list of ordered metrics will allow choosing the optimal number of routers for node v:

$$l[S_{R*(v)}(v)] \le [S_{R*(v)(v)}]. \quad (17)$$

B. The Algorithm for the Choice of the Path

The basis for the classic routing algorithm is considered by the breadth-search procedure, which allows analysing the graph *G* (*V*, *E*) in width. Realising the procedure for each network node, two variables *dist*(*v*) and *pred*(*v*) [3] are added. There is a definite starting value for variable $dist(v) \to \infty$. This means that the node has not been visited yet. The coordinator node d possesses a variable value $dist(v) = 0$. The initial value of the variable – $pred(v) = 0$, it means that the node is without a router.

The procedure $BS(G, d)$ ensures testing all the nodes and giving value to variable *dist*(*v*) of each node; the value is equal to the length of the path from node v to node d. For the validation of the data, a row *Q* has been used with the function ***First_OutQ***(*Q*) (excluding the first element in the row) and ***Insert_End*(*Q*)** (inserting the element at the end of the row).

```
Procedure BS(G,d)
for all u ∈ V { dist(u):=∞; pred(u)=0;}
dist(d)=0;
Q=[d]
while Q is not empty {
u=Fist_Out(Q);
for all(v,u) ∈ E {
if (dist(v)=∞) {dist(v)=dist(u)+1; pred(v)=u;
Insrt_End(Q);}
}
}
```

Carrying out the cycle, the variable value *dist*(*v*) is the minimum distance from the node v to d, but values of *pred*(*v*) will allow creating the path from any of vertices v to coordinator node d. As a result, the given approach can be considered the optimal one in the tasks where the graph path metrics are used.

Using the defined mathematical model of the network, it is necessary to introduce an approach that will list and take into account the remaining amount of energy in each of the network nodes. It is known that the amount of energy left in the node is a positive figure. In this case, it is proposed to use Dijkstra's algorithm (Fig. 4) to analyse the graph under the condition $l(u, v) \ge 0$ – where $l(u, v) = E_0 - E_n$ as a result we have to build the path with a minimum energy difference coefficient.

```
Procedure Deikstra(G,l,d)
for all u ∈ V { dist(u):=∞; pred(u)=0;}
dist(d)=0; H=CreateQueue(V);
while H is not empty {
u=Fist_Out(H);
for all(v,u) ∈ E {
if (dist(v)=dist(u)+l(v,u))
   {  dist(v)=dist(u)+l(v,u);     pred(v)=u;
Rearrange_Queue(H,v);}
}
}
```

The significant difference in the procedure is the use of priority rows *H* where all the elements are sorted using *dist*(*v*) values. The procedure ***CreateQueue*()** forms the priority row *H* placing the coordinator node *dist*(*d*) = 0 at the start of the row. The procedure ***Rearrange_Queue*(*H*, *v*)** ensures the reconfiguration of the row in case of change of metrics v.

Every time executing ***while*** cycle, two conditions are met:

1. There is a *d* > 0 value, where all nodes are accessed – $dist(v) \le d$ and all the remaining nodes in the row – $dist(v) \ge d$.
2. The value *dist*(*v*) in the row of each node v is equal to ∞ or the minimum path from v to d. In addition, unused nodes are removed from the row.

The fulfilment of conditions guarantees the optimal results of the algorithm solution.

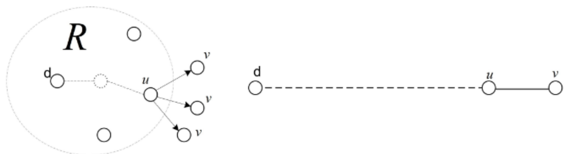

Fig. 4. Applying Dijkstra's algorithm.

If there is a necessity to assess the changes in the metrics of the graph vertices, the Bellman-Ford algorithm is proposed to be used. While operating, the initial value of metrics for node v may frequently change. Every time it changes, it is realised through the procedure:

$$Update\_M(v,u) : dist(v) - min\{dist(v), dist(u)\_l(v,u)\}. \quad (18)$$

After performing the procedure, the certain characteristics are carried out:

- The use of the procedure does not increase the values of *dist*(*v*);
- An optimum result corresponds to the value of minimum *dist*(*v*), as a result, the regular procedures Update_M will not affect the results negatively.

```
Procedure Bellman_Ford(G,l,d)
for all u∊ V { dist(u):=∞; pred(u)=0;}
dist(d)=0; H=CreateQueue(V);
  repeat (|V |-1) times {
    for all e∊  E Update_M(e)
}
```

To make the described algorithms applicable in wireless sensor networks, it is necessary to carry out a small adoption of each algorithm. To improve Dijkstra's algorithm, it is proposed to introduce additional function Best_Value (Q) that allows from the cluster *Q* distributing node u with a minimum path metric on d. Node cluster *Q* not examined at the initiation stage is considered to be the whole cluster of nodes V, the coordinator node d is given metric 0.

```
Procedure Deikstra_updated(G,d)
  Q ← V
  while Q≠0 {
   u ← Best_Value(Q)
  for each v ∊ N(u) ∩ Q do {
        if l(S(v) ⊕ S(u)) ≺ l(S(v))
  S(v) ← S(v) ⊕ S(u)
     }
  }
```

Concluding the algorithm, for each of $v \in V$ nodes the path *S*(*v*, *d*) is created for the transfer of information to node d. The algorithm is repeated as many times as there are coordinator nodes or as long as all the nodes are not used.

Improving the Bellman-Ford algorithm, let us introduce the function *FindRelay*(), which in case of necessity withdraws the existing list v of the transponder and creates a new one by inserting the nodes from *N*(*v*) list of metrics in reducing order until the next connection of the router allows reducing the value of metric pertaining to node *l*(*S*(*v*)).

```
Procedure Bellman_updated(G,d)
  Initialize : l(S(vi )) = ∞; l(S(d)) = 0;
  repeat {
   sorting of nods DESC;
  for i=2 to (|V | - 1) do {
    (S(vi )) = FindRelays(v)
    }
  } untill not identification changes.
```

where:

```
Function  FindRelays(v)
  Router sorting DESC;
Initialize : S(v) = 0; j = 1;
  while l[S(v) ⊕ S(uj )] ≺ l[S(v)] {
       S(v) = S(v) ⊕ S(uj );
  j = j + 1;
     }
FindRelays :=  l(S(v))
```

The proposed algorithms constitute the optimal path for all the network nodes if for ((*v*)) metrics: the remaining amount of energy E_0 in the node is available. If it is not possible to identify the value of E_0 – the network is no longer able to work.

VI. Conclusion

The proposed reconfiguration method of the network makes it possible to increase the life cycle of wireless sensor network by smoothing the remaining amount of energy in the network. It is known that the total amount of energy consumed in the node depends on many factors [3], [8]:

- technical solutions of hardware;
- frequency of operation;
- intensity of data flow;
- environmentally accessible algorithms.

The article describes the developed method that increases the life cycle of the wireless sensor networks; this method allows equalising the amount of energy consumed in each of the network nodes and preventing the overload during the information transmission. This approach is different from others offered because:

- it uses dynamically configurable agent in each of the network nodes;
- it uses BST customised optimal data flow path algorithm;
- it uses the coefficient of the change of network configuration and sets the time for the denial of the system.

The developed method described in the article allows effectively planning the structure of the network dividing the tasks for network nodes, thus increasing the overall life cycle of the network.

References

[1] S. Basagni, A. Carosi, C. Petrioli, "Controlled Vs. Uncontrolled Mobility in Wireless Sensor Networks: Some Performance Insights," *Vehicular Technology Conference*, 2007. VTC-2007 Fall. 2007 IEEE 66th. 2007. pp. 269–273. http://dx.doi.org/10.1109/vetecf.2007.70

[2] D.M. Blough, P. Santi, "Investigating upper bounds on network lifetime extension for cell-based energy conservation techniques in stationary ad hoc networks," *Proc. of the 8th annual int. conf. on Mobile computing and networking. MobiCom '02*. New York, NY, USA: ACM, 2002. pp. 183–192. http://dx.doi.org/10.1145/570645.570668

[3] Y. Chen, Q. Zhao, "On the lifetime of wireless sensor networks," *Communications Letters, IEEE*. Nov., vol. 9, no. 11. pp. 976–978.

[4] C. Fok, G. Roman, C. Lu. "Rapid Development and Flexible Deployment of Adaptive Wireless Sensor Network Applications," *Proc. of 25th IEEE Int. Conf. on Distributed Computing Systems*, 2005, pp. 653–662.

[5] D. Georgoulas, K. Blow, “Making Motes Intelligent: An Agent-Based Approach to Wireless Sensor Networks,” *The World Scientific and Engineering Academy and Society (WSEAS) on Communications Journal*, 2006, pp. 515–522

[6] S. Halder, A. Ghosal, A. Chaudhuri, S. DasBit, “A probability density function for energy-balanced lifetime-enhancing node deployment in WSN,” *Proc. of the 2011 int. conf. on Computational science and its applications*, vol. 6785 Part IV. ICCSA’11. Berlin, Heidelberg: Springer-Verlag, 2011. pp. 472–487. http://dx.doi.org/10.1007/978-3-642-21898-9_40

[7] I.S. Hammoodi, B. Stewart, A. Kocian, S. McMeekin, “A Comprehensive Performance Study of OPNET Modeler for ZigBee Wireless Sensor Networks,” *Next Generation Mobile Applications, Services and Technologies, 2009. NGMAST '09. 3rd Int. Conf.* 2009. pp. 357–362. http://dx.doi.org/10.1109/ngmast.2009.12

[8] A. Jurenoks, L. Novickis, “Wireless Sensor Networks LifeTime Assessment Model Development,” In: *Proc. of 10th Int. Scientific and Practical Conf. “Environment. Technology. Resources”*, Rezekne, Latvia, 2015. June, ISSN 1691-5402 (SCOPUS), pp. 121–126.

[9] S. Kumar, A. Arora, T. Lai, “On the lifetime analysis of always-on wireless sensor network applications,” *IEEE International Conference on Mobile Adhoc and Sensor Systems Conference*, 2005. pp. 3–188. http://dx.doi.org/10.1109/MAHSS.2005.1542797

[10] R. Mautz, S. Tilch, “Survey of optical indoor positioning systems,” *Int. Conf. on Indoor Positioning and Indoor Navigation*, IPIN, 2011. pp. 1–7. http://dx.doi.org/10.1109/ipin.2011.6071925

[11] K. Mohammadi, H. Hamidi, “Evaluation of fault-tolerant mobile agents in distributed systems,” *Proc. of the first IEEE and IFIP Int. Conf. in central Asia on Internet*, 2005. http://dx.doi.org/10.1109/canet.2005.1598208

[12] H. Qi, S.S. Iyengar, K. Chakrabarty, “Multiresolution Data Integration Using Mobile Agents in Distributed Sensor Networks,” *IEEE Transactions on Systems, Man and Cybernetics Part C*, 2001, pp. 383–391.

[13] J.-Z. Sun, J. Sauvola, “Mobility and mobility management: a conceptual framework,” *10th IEEE International Conference*, Networks, 2002. ICON 2002, 2002. pp. 205–210.

[14] L. Zhuang, W. Liu, J.-B. Zhang et al. “Distributed asset tracking using wireless sensor network,” *IEEE Int. Conf. on Emerging Technologies and Factory Automation*, 2008. ETFA 2008. pp. 1165–1168. http://dx.doi.org/10.1109/etfa.2008.4638546

[15] F.P. Vasilyev, Chislennyye metody resheniya ekstremalnykh zadach: Ucheb. posobiye dlya vuzov. 2-e izd., pererab. i dop. M.: Nauka, 1988. 552 s. (In Russian).

[16] R.O. Kurpatov, Issledovaniye i razrabotka energoeffektivnogo metoda lokalizatsii elementov besprovodnykh sensornykh setey: dis. kand. tekhn. nauk: 05.12.13. M., 2011. 126 s. (In Russian)

[17] V.YU Yurochkin., T.I. Mokhseni, Iyerarkhicheskiye podkhody k samoorganizatsii v besprovodnykh shirokopolosnykh sensornykh setyakh na osnove khaoticheskikh radioimpulsov, Trudy MFTI. 2012. T. 4, no. 3. s. 151–161. (In Russian).

Aleksejs Jurenoks is a Lecturer at Riga Technical University. He obtained *Mg. sc. ing.* degree in 2004. He is the author of 12 publications and 6 educational books.
He is regularly involved in different EU-funded projects: eLOGMAR-M (FP6, 2004–2006); IST4Balt (FP6, 2004–2007), UNITE (FP6, 2006–2008) and BONITA (INTERREG, 2008–2012).
E-mail: Aleksejs.Jurenoks@rtu.lv

Leonids Novickis is a Head of the Division of Software Engineering at Riga Technical University. He obtained *Dr. sc. ing.* degree in 2004 and *Dr. habil. sc. ing.* degree in 1990 from the Latvian Academy of Sciences.
He is the author of 180 publications. Since 1994, he is regularly involved in different EU-funded projects: AMCAI (INCO COPERNICUS, 1995–1997) – WP leader; DAMAC-HP (INCO2, 1998–2000), BALTPORTS-IT (FP5, 2001–2003), eLOGMAR-M (FP6, 2004–2006) – scientific coordinator; IST4Balt (FP6, 2004–2007), UNITE (FP6, 2006–2008) and BONITA (INTERREG, 2008–2012) – RTU coordinator; LOGIS, LOGIS-Mobile and SocSimNet (Leonardo da Vinci) – partner, eINTERASIA (FP7, 2013–2015) – project coordinator. He was an independent expert of IST and Research for SMEs in FP6 and FP7. He is a corresponding member of the Latvian Academy of Sciences and an elected expert of the Latvian Council of Science.
E-mail: Leonids.Novickis@rtu.lv