Modeling and Optimizing of Connections for Dynamic Sensor Fields Based on BT-Graph

Tuyen Phong Truong[1(✉)], Huong Hoang Luong[2], Hung Huu Huynh[3], Hiep Xuan Huynh[2], and Bernard Pottier[1]

[1] Lab-STICC, UMR CNRS 6285, Université de Bretagne Occidentale, Brest, France
{phong-tuyen.truong,pottier}@univ-brest.fr
[2] Cantho University, Can Tho, Vietnam
{lhhuong,hxhiep}@ctu.edu.vn
[3] Danang University of Technology, Danang, Vietnam
hhhung@dut.udn.vn

Abstract. In this paper, we propose a new approach to model and optimize the dynamic sensor field for both internal network connections and LEO satellite connection based on BT-Graph. Due to the shift of LEO satellite's orbit at each revolution, a dynamic sensor field (DSF), which is able to redetermine its gateways, is suitable to improve successful data communications. It is convenient to present a DSF as a BT-Graph that aims to utilize optimization algorithms. The simulation experiments are performed on a forest fire surveillance network to validate our proposed approach.

Keywords: Associate analysis · Balltree structure · BT-Graph · Dynamic sensor field · Leo satellite

1 Introduction

Wireless Sensor Network (WSN) [1] is known as a network of sensors cooperatively operating in order to surveillance or collect the environmental parameters. Although the short transmission range can be compensated by applying a mesh topology, it would be economically infeasible to deploy in large geographic areas or behind obstacles (mountains, oceans, ...). To overcome these disadvantages, satellite-based wireless sensor networks have emerged as a promising technology with various applications.

Because the orbit of a LEO satellite shifts in westward direction around the polar axis at each revolution [10], the meeting points of a gateway on the Earth's surface and the LEO satellite will be changed over time. With a static sensor field, it can be occasionally unsuccessful in communication with the LEO satellite because the meeting time is not enough for data exchange [10]. To overcome this problem, a dynamic sensor field (DSF) [1], which has the ability to redetermine its gateway to adapt with the shifts of LEO satellite's paths, is suitable to improve the connection time.

P.C. Vinh et al. (Eds.): ICTCC 2016, LNICST 168, pp. 297–310, 2016.
DOI: 10.1007/978-3-319-46909-6_27

In order to improve the connection time, it is necessary to choose proper gateways for the longest length of connection time. For this purpose, the connections between a LEO satellite and a dynamic sensor field should be presented by a graph-based model because it is convenient to determine number of neighbors in the satellite's communication range and then apply the optimization algorithms [3,5]. In addition, BT-Grap model [2] based on balltree structure is efficiently support not only for searching the range nearest neighbors (RNN) but also finding the shortest path to a given target node [6,13,15,20]. In this paper, we propose a new approach, namely dynamic sensor field optimization model based on BT-Graph (BT-DYNSEN), to model and optimize the DSF in communication with LEO satellite.

The rest of this paper is organized as follows. In Sect. 2, we overview related works. Section 3 presents the graph-based model for a DSF based on BT-Graph (BT-DYNSEN). How to optimize the DSF for satellite connection is presented in Sect. 4. Section 5 gives the simulation experiments on a forest fire surveillance network in Vietnam before a conclusion is drawn.

2 Related Work

The LEO satellite communication can be used in mobile satellite communications, surveillance the Earth surface, geological surveys, so on [1,11,14]. In the last decade, researches on communication services provided by LEO satellites have focused on several main directions such as optimizing the mechanics, interconnections, electric circuits, power supply. Another approach, many surveys show the research topics in design trajectory, handover traffic and constellation, as well as design protocols, radio frequencies, onboard transceivers and antenna designs [1,14]. However, the direct radio links between sensor fields and LEO satellites are not considered in literature. In recent years, it emerges as an attractive topic because of the current innovation solutions such as LoRa Semtech and solutions from vendors QB50 [9].

Additionally, graph-based model has emerged as well approach to present the structure and elaborate the performance of wireless sensor networks [3,4]. For example, random geometric graph [15] was used to determine the probability of the whole network being connected. Secure communications between large number of sensor nodes in WSNs can be elaborated on expander graph [19] and finding transmission path in network was performed based on Pascal graph [5,7]. Furthermore, hyper-graph [18] was utilized to support for reducing the transmission energy consumption and improving the fault-tolerant ability of the system. In the next section, we introduce BT-DYNSEN model for optimizing the DSF for the connection with LEO satellite.

3 BT-DYNSEN

3.1 BT-DYNSEN Model of DSF

BT-DYNSEN model of a dynamic sensor field (DSF) [1] based on BT-Graph [2] is a graph G(V,E). In this graph, set of vertices $V = \{v_i\}$, $i = 1..n$ corresponds to sensor nodes and set of edges, $E = \{e_j\}$, $j = 1..m$ are connections between the nodes with associated weight functions $W = \{w_j\}$, $j = 1..m$. The value of each w_j is given by Euclidean distance $d(v_i, v_j)$, $i \neq j$. Additionally, $R = \{r_i\}$, $i = 1..n$ are the radii of the communication ranges of nodes [2,6,8]. An edge is established if and only if the distance between two nodes is less or equal to the minimum value of their communication radii, $d(v_i, v_j) \leq \min(r_i, r_j)$, $i \neq j$. Note that the terms *node* and *vertex* are used interchangeably in this paper as a matter of convenience.

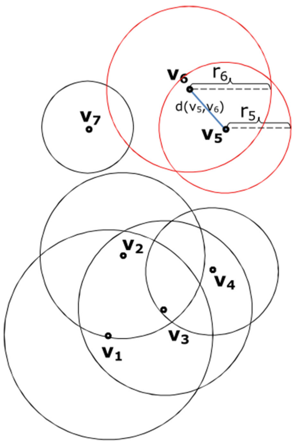

Fig. 1. A dynamic sensor field in which the communication ranges of sensor nodes are indicated by the radii of balls.

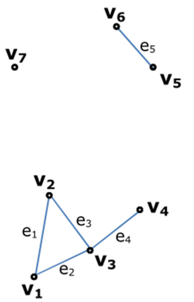

Fig. 2. A BT-Graph of the dynamic sensor field with 07 vertices $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$ and 5 edges $E = \{e_1 = e(v_1, v_2), e_2 = e(v_1, v_3), e_3 = e(v_2, v_3), e_4 = e(v_3, v_4), e_5 = e(v_5, v_6)\}$.

In Fig. 1, for an example, a pair of vertices (v_5, v_6) has communication ranges r_5 and r_6 respectively. Because the distance between v_5 and v_6 is less than r, $d(v_5, v_6) < r$, so there exists an edge e_5 connecting them as can be seen in Fig. 2. Similarly, the others edges of this graph namely $e_1 = e(v_1, v_2), e_2 = e(v_1, v_3), e_3 = e(v_2, v_3), e_4 = e(v_3, v_4)$ could be established. There are $2^n - 5$ edges between a pair of nodes of this graph that are not existed due to inadequacy of the condition. The vertex v_7 is isolated because all distance values between it and the other nodes are inadequate to the condition.

3.2 BT-DYNSEN Model of LEO Satellite Connection

The connections between a LEO satellite and a dynamic sensor field are also described in a BT-Graph. Let $V = \{v_i\}$, i=1..n, is a set of sensor node coordinates in a DSF. In this scenario, connection time is defined when any sensorset [1] of the

DSF under the satellite coverage. The LEO satellite communication range is considered as a circle whose center is sub-point on the ground (sub-satellite point), s. It is noted that sub-satellite point, s, is where on the ground the straight line connecting the center of the Earth and the satellite meets the Earths surface [10]. The associated BT-Graph consists of $n+1$ vertices $P = \{V, s\} = \{v_1, v_2, ..., v_n, s\}$. If a node is within the communication range of the satellite, there exists an edge with weight given by the Euclidean distance between them. Otherwise edge weight is set to infinity. Consequently, the number edges of the graph are $m + n$ by adding n new edges $C = \{c_1, c_2, ..., c_n\}$ with $c_i = c(s, v_i)$, $i = 1..n$. The weights of the n new edges are denoted by $Z = \{z_1, z_2, ..., z_n\}$. In this case the set of edges is $R = \{E, C\} = \{e_1, e_2, ..., e_m, c_1, c_2, ..., c_n\}$ and the set of corresponding weights is $Q = \{W, Z\} = \{\{w_k\}, \{z_i\}\}$ with $k = 1..m$, $i = 1..n$. Hence the BT-DYNSEN model for LEO satellite connections is presented by a graph G(P,R) with weight functions Q.

Furthermore, during the connection time only one sensor node (vertex) of a dynamic sensor field is chosen to connect with the sub-satellite point (center vertex) at one time [1]. A number of different nodes could be chosen based on the value of edge weights at different times. To manage the connections, the name of chosen node is kept in *Connection vector* [1] and the corresponding time is saved in *Time vector* [1] (as in Fig. 3).

Connection number	1	2	3	...
Connection vector	v_1	v_3	v_4	...
Time vector	t_1	t_2	t_3	...

Fig. 3. An example of satellite connection data with three rows: *Connection number*, *Connection vector* and *Time vector*.

Figure 3 shows that in *connection 1*, center vertex s connects with v_1 at time t_1. In a similar way, in *connection 2* at time t_2 and *connection 3* at time t_3, v_4 and v_2 are chosen to connect with s respectively.

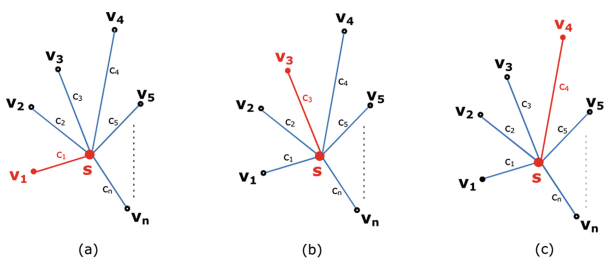

Fig. 4. The BT-Graph of a dynamic sensor field in three different connections (solid red lines) at times t_1, t_2 and t_3. (Color figure online)

For instance, Fig. 4 shows three graphs of the dynamic sensor field in three different connections with the center vertex s (a sub-satellite point) at different times t_1, t_2 and t_3. At time t_1 (see Fig. 4(a)), vertex v_1 is chosen and edge c_1 is established. Similarly, in Fig. 4(b) and Fig. 4(c) vertex v_3, v_4 are chosen for connections that leads to corresponding edges c_3, c_4 are established at time t_2 and t_3 respectively.

3.3 Compute the Connection Time

In this section, we describe the way to calculate connection time, t_{ij}, between a LEO satellite and a gateway of DSF, v_j [1,10]. Similarly, the calculation could be applied for all other nodes. Note that every node of the sensor field is assumed as a gateway for the connection with the satellite in calculating the values of connection time.

First, it is necessary to define the angles and related distances between satellite, a gateway on the ground and the Earth's center. The parameters are indicated on Fig. 5. For angular radius of the spherical Earth, ρ_i, and the angular radius λ_{0_i} can be found from relations

$$\sin(\rho_i) = cos(\lambda_{0_i}) = \frac{R_E}{R_E + H} \tag{1}$$

$$\rho_i + \lambda_{0_i} = 90\ deg \tag{2}$$

where $R_E = 6378.14$ km is the Earth's radius and H is the altitude of the satellite above the Earth's surface.

With the coordinates of a sub-satellite point, s_i $(Long_{s_i}, Lat_{s_i})$ along each satellite's ground track and a sensor node, nodes v_j of the DSF $(Long_{v_j}, Lat_{v_j})$, and defining $\Delta L_{ij} = |Long_{s_i} - Long_{v_j}|$, the azimuth, $\Phi_{E_{ij}}$, measured eastward from north, and angular distance, λ_{ij}, from the sub-satellite point to the sensor node (see Fig. 6) are given by

$$cos(\lambda_{ij}) = sin(Lat_{s_i})sin(Lat_{v_j}) + cos(Lat_{s_i})cos(Lat_{v_j})cos(\Delta L_{ij}) \tag{3}$$

$$cos(\Phi_{E_{ij}}) = ((sin(Lat_{v_i}) - cos(\lambda_{ij})//sin(Lat_{s_i}))/(sin(\lambda_{ij})cos(Lat_{s_i})) \tag{4}$$

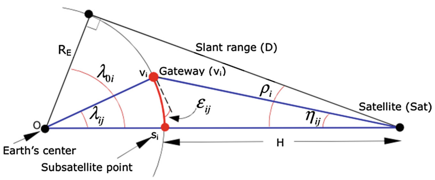

Fig. 5. Gateway of sensor field geometry [10].

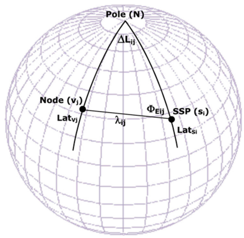

Fig. 6. Relationship between sensor node of a DSF and sub-satellite point [10].

where $\Phi_{E_{ij}} < 180\ deg$ if v_i is east of s_i and $\Phi_{E_{ij}} > 180\ deg$ if v_i is west of s_i Consider triangle Os_iv_j (in Fig. 5), the distance, d_{ij}, between s_i and v_j can be found using the law of cosines:

$$d_{ij}^2 = R_E^2(1 - cos(\lambda_{ij})) \tag{5}$$

The *connection time*, t_{ij}, is given by

$$t_{ij} = (\frac{P}{180\ deg}) \arccos(\frac{\cos(\lambda_{ij_{max}})}{\cos(\lambda_{ij_{min}})}) \tag{6}$$

where P is the *orbit period* in minutes. From Eqs. (5) and (6), the communication duration, t_{ij}, strongly depends on how close the nodes v_j is to the sub-satellite points s_i, the distances d_{ij}, along the ground track on any given orbit pass [10].

4 Optimization Method

4.1 Verify the Connectivity of DSF

In this work, top down construction algorithm is chosen in order to build balltree structure for verifying the network connectivity of the DSF. In this manner, the time complexity can be found in $O(nlog^2n)$ [8].

Algorithm 1. verifyConnectivity(D)

```
if D.hasAPoint() then
    create a leaf B is a point in D;
    return B;
else
    let c is the dimension of greatest spread;
    let L, R are the set of points lying to 2 subsets;
    let r is a radius of ball;
    create B with two children:
        B.center ← c;
        B.radius ← r;
        B.leftChild ← balltreeConstruct(L);
        B.rightChild ← balltreeConstruct(R);
return B;
```

Top down algorithm to construct the balltree is a recursive process from top to down. The process at each step is to choose the split dimension it and then split the set of values into two subsets. First, it is necessary to choose the point in the ball which is farthest from its center, and then choose a second point (p_2)

that is farthest from (p_1). It is followed by assigning all points in the ball to closest one of two clusters corresponding to p_1 and p_2. Finally, the centroid of each cluster is calculated and the minimum cluster radius for enclosing all the points are determined. This process is stopped only when the leaf balls contain just two points. Note that D denotes current considered balltree structure and B denotes leaf node defined after each process.

4.2 Determine the Sensorsets Corresponding to Each Sub-satellite Point

Defining a sensorset [1] based on BT-DYNSEN is to find k nearest sensor nodes (neighbors) under the satellite's coverage area at each sub-satellite. It is a depth-first traversal algorithm for traversing tree or graph data structures, starting with the root node. The value of Q is updated during search process. At each considered node B, Q obtains k points which are nearest query q as following algorithm.

Algorithm 2. determineSensorset

```
input: balltreeNode, query
output: sensorset
begin
    centerNode ← balltreeNode.getCenter();
    radiusNode ← balltreeNode.getRadius();
    //calculate distance from query to centerNode
    d ← distance(centerNode.lat, centerNode.lon, query.lat, query.lon);
    if(balltreeNode.isALeaf) {
        if(d < query.range) {
            write name of balltreeNode to file;
        }
    }
    else {
        if(d < (query.range + radiusNode)) {
            left ← balltreeNode.leftChild;
            right ← balltreeNode.rightChild;
            rnn: left, query;
            rnn: right, query;
        }
    }
```

There are two different cases in *BT-DYNSEN* algorithm as follows. If current considered node is a leaf node and the distance from query point q to B is less than r ($d < r$), the obtained result is updated by adding B into Q. Otherwise, if B is not a leaf node and the distance from query point q to B is less than the total of r and the radius of B ($d < r + B.radius$), it is necessary to perform recursive algorithm for the two child nodes of a parent node B: left-child and right-child.

4.3 Select the Gateways of DSF

If a DSF V has n nodes, there are $2^n - 1$ connection items, $G = \{g_l\}$, l=1..($2^n - 1$). It is necessary to find out a set of proper nodes which play as gateways of DSF to provide the longest length of time for the connection. To do this, the *association analysis algorithm* [13] is applied. To do this, a DSF is represented in a binary format, where each row corresponds to a connection item and each column corresponds to a node. A node value is one if the node appears in a connection item and zero otherwise. Weight of a connection item determines how often a connection item is applicable to a set of connection items. The weight of a connection item, $g_i \in G$, can be defined as $w(g_i) = \mid \{g_l \mid g_i \subseteq g_l, g_l \in G\} \mid$, where the symbol | . | denotes the number of elements in a set.

The algorithm for selecting the gateways of the DSF is briefly presented in following list.

Algorithm 3. selectGateway()

```
input: list of sensorsets
output: gateways of DSF
//C_k: candidate sensors (size k)
//L_k: set of selected gateways (size k)
begin
    L_1 ← {weight1 − sensorsets};
    for(k = 2; Lk.count > 0; k++) {
        C_{k+1} ← generate candidate sensors L_k;
        foreach (transaction t in data) {
            Increment count of the candidate sensors in C_{k+1} that contained
            in t;
        }
        L_{k+1} ← candidate sensors in C_{k+1};
    }
    Write counted frequent of all sensors L_k to file;
end
```

For example, a DSF with 4 nodes, $V = \{v_1, v_2, v_3, v_4\}$ is shown in Fig. 7. There are 4 sensorsets $A_1 = \{v_1\}$, $A_2 = \{v_1, v_2, v_3\}$, $A_3 = \{v_2, v_3, v_4\}$ and $A_4 = \{v_3, v_4\}$. The weights of connection items are presented in Fig. 8. The connection with the highest weight corresponds to the least number of times required to change the gateways. It leads to the connection duration time of the connection is longest. Because the weight of connection item (v_1, v_3) is highest, this connection is chosen.

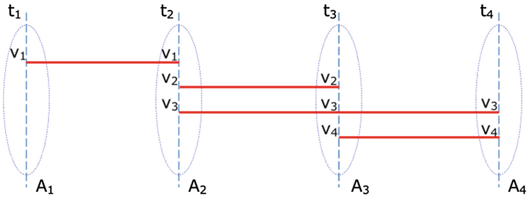

Fig. 7. The connections between a 4-node DSF with a LEO satellite.

Connection items	Weights
(v_1, v_2, v_4)	1
(v_1, v_2, v_3)	1
(v_1, v_3)	3
(v_1, v_3, v_4)	1

Fig. 8. The weights of connection items in a DSF with 4 nodes.

4.4 Find Shortest Path for Data Dissemination

The problem of finding shortest path from each sensor node of DSF to the gateway can be solved by a graph search method. The algorithm is presented as follows:

Algorithm 4. searchShortestPath()

```
Init matrix M;
S ← Start point;
T ← End point;
d is an array;
free is an array;
trace is as array;
d[S] ← 0;
    while (true){
        u ← -1;
        min ← INFINITE;
        for (i ← 1 to n-1) {
            if (free[i] AND (d[i] < min)) {
               min ← d[i];
               u ← i;
            }
        }
        if((u = −1) OR (u = T))
            Break;
        free[u] ← false;
        for (v ← 0 to n-1) {
            if(free[v] AND (d[v] > (d[u] + M[u, v]))) {
            d[v] ← d[u] = arr[u][v];
            trace[v] ← u;
        }
    }
}
```

The algorithm proceeds in three following steps. First, the balltree graph-based model, G, is constructed. At the next step, the weight matrix M for edges connect each pair of vertices (sensor nodes) in V is computed based on their coordinates. For v_i, v_j are two different vertices in V, in case of $v_i \equiv v_j$ the edge weight is 0. If $v_i \not\equiv v_j$, the edge weight is given by $d(v_j, v_j)$ if $d(v_j, v_j) \leq r$, otherwise it is infinity, ∞. Finally, the shortest path from q to e in the weight matrix is figured out. Note that q is starting point (a sensor node) and e is the destination point (the gateway of DSF).

5 Experiment

5.1 Data Used

For experiments, an abstract structure of the long-range sensor field for fire forest surveillance was generated by using NetGen [16]. Figure 9(a) shows dynamic sensor field consists of 50 sensor nodes that is stretched from South Central Coastal to Southeast and extended up to Mekong River Delta in Vietnam.

From the constraints about the satellite's orbit altitude in [1], orbit 12974 of BEESAT-3 [17], a LEO satellite with orbit altitude is around 575 km, which was chosen in our experiments. The ground track data stored in a plain text (.txt file) was used as input data.

5.2 BT-DYNSEN Tool

We have developed the BT-DYNSEN tool in MATLAB, that enables to optimize the connection time between a dynamic sensor field with a LEO satellite. GPredict [12] is used to provide the information about BEESAT-3 satellite's path. Besides, NetGen tool [16] is utilized to generate the abstract network of a 50-node dynamic sensor field from geographic data provided by Google maps service. The obtained result are the nodes of the DSF which should be configured as gateways for the best connection duration time and the shortest paths for data dissemination from each node to these gateways.

5.3 Experiment 1: Verify Network Connectivity

The connectivity of the 50-node dynamic sensor network was verified by applying BT-DYNSEN as shown in Fig. 9(a). Figure 9(b) depicts the balltree structure of this DSF in which there are 49 balls with radii from 10.124 m to 321.165 m. BT-Graph model then was utilized to ensure the full connectivity of all network nodes, the radius 30.497 m was then chosen as shown in Fig. 9(c).

5.4 Experiment 2: Determine the Sensorsets

BT-DYNSEN tool was employed in determining sensorsets corresponding to sub-satellite points during visiting time. The map in Fig. 10 shows the sensorset was

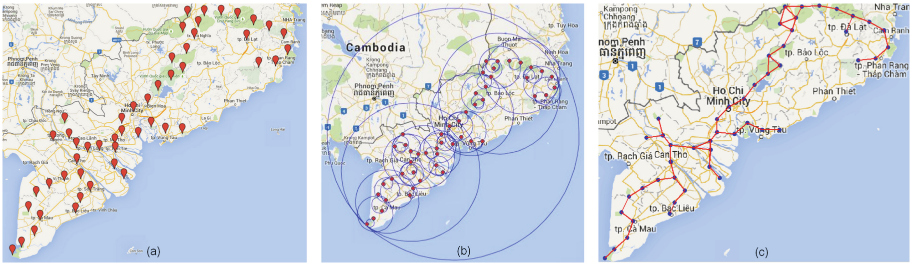

Fig. 9. Verify dynamic sensor network connectivity. (a) A 50-node dynamic sensor network, (b) Balltree structure of the DSF, (c) Connectivity of the DSF.

Fig. 10. Determine the sensorsets corresponding to each sub-satellite along the ground track of BEESAT-3 in orbit 12794. (Color figure online)

determined with sub-satellites of BEESAT-3 at (latitude: 14.00, longitude: 102.48) in orbit 12794. Sub-satellite and satellite coverage were indicated by solid red square and red circle respectively. The sensorset consists of 26 sensor nodes which were under the satellite's coverage area as indicated by a solid red circle. There are four sensorsets were created along the BEESAT-3's ground track in orbit 12794.

5.5 Experiment 3: Select the Gateways

With each sensorset, a subset of connections is established. The weights of each connection in subset are then computed. A set of connection items is created by combining these subsets. The best connection is chosen based on the weights of connection items. For instance, in Fig. 11 node v_{36} was chosen as the gateway of the 50-node DSF to connection with BEESAT-3 satellite in orbit 12794 because its weight is highest in sensor nodes.

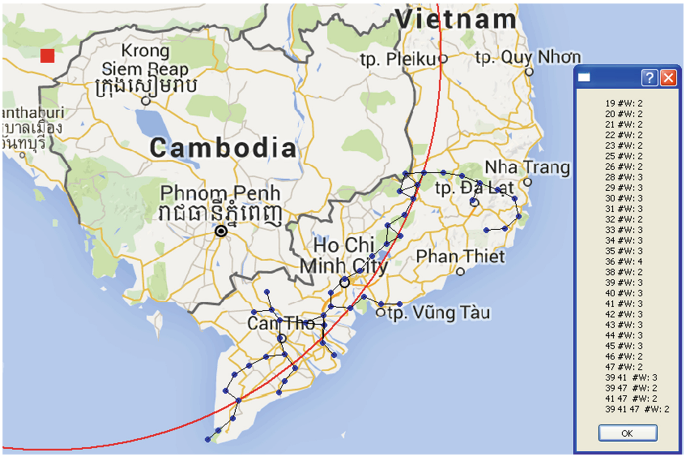

Fig. 11. Select a set of nodes which play as gateways of the DSF according to the weights of connections.

5.6 Experiment 4: Optimize Data Dissemination

To ensure sensing data to be collected from all sensor nodes and then sent before the satellite leaving, it is necessary to find the shortest path from each sensor nodes to the gateway. The interconnection weights within the DSF are geographic distances between each pair of nodes that were carried out by applying BT-DYNSEN model. Figure 12, as an example, illustrates the chosen path (the bold blue line) for data dissemination from sensor node v_{50} to the gateway node v_{36}.

Fig. 12. The shortest path for data dissemination from node v_{50} to gateway node v_{36}. (Color figure online)

6 Conclusion

Based on BT-Graph, we have described a new approach in order to model and optimize the dynamic sensor field for LEO satellite connections. The distances between the sub-satellite points and each node of the sensor field is utilized as a key factor to find out the proper gateways for the longest connection time. The experimental results were obtained by applying several appreciate algorithms on BT-DYNSEN model to verify the connectivity of network, determine sensorsets at visiting time, choose set of gateway nodes and find shortest path for data dissemination in DSF. The simulation results show that our proposed graph-based model helps to increase the amount of time for data communications in long-range sensor field applications using satellite connections in order to monitor, control and collect environmental data.

Acknowledgment. The first author was funded by the program 911 of Ministry of Education and Training, Vietnam. This publication was made possible through support provided by the Lab-STICC, UMR CNRS 6285, France.

References

1. Truong, T.P., Van Tran, H., Huynh, H.X., Pottier, B.: Optimizing the connection time for LEO satellite based on dynamic sensor field. In: Vinh, P.C., Alagar, V. (eds.) Context-Aware Systems and Applications. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 165, pp. 380–394. Springer, Heidelberg (2016)
2. Luong, H.H., Huynh, H.X.: Graph-based model for geographic coordinate search based on balltree structure. In: Proceeding of the 17^{th} International Conference, Daklak, Vietnam, pp. 116–123 (2014)
3. Akoglu, L., Tong, H., Koutra, D.: Graph based anomaly detection and description: a survey. Data Min. Knowl. Discov. **29**(3), 626–688 (2015). IEEE Press
4. Khou, I., Minet, P., Laouiti, A., Mahfoudh, S.: Survey of deployment algorithms in wireless sensor networks: coverage and connectivity issues and challenges. Int. J. Auton. Adapt. Commun. Syst. (IJAACS), p. 24 (2014)
5. Chaparro, C., Eberle, W.: Detecting anomalies in mobile telecommunication networks using a graph based approach. In: FLAIRS Conference, pp. 410–415 (2015)
6. Abbasifard, M.R., Ghahremani, B., Naderi, H.: A survey on nearest neighbor search methods. Int. J. Comput. Appl. 39–52 (2014)
7. Panwar, D., Neogi, S.G.: Design of energy efficient routing algorithm for wireless sensor network (WSN) using Pascal graph. Comput. Sci. Inf. Technol. 175–189 (2013)
8. Omohundro, S.M.: Five Balltree Construction Algorithms. Technical report (1989)
9. Lucas, P.-Y., Van Truong, N.H., Truong, T.P., Pottier, B.: Wireless sensor networks and satellite simulation. In: Pillai, P., Hu, Y.F., Otung, I., Giambene, G. (eds.) Wireless and Satellite Systems. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 154. Springer, Heidelberg (2015)
10. Larson, W.J., Wertz, J.R.: Space mission geometry. In: Space Mission Analysis and Design, Chap. 5, 3rd edn. pp. 95–230. Microcosm Press, Portland (2003)

11. Colitti, W., Steenhaut, K., Descouvemont, N., Dunkels, A.: Satellite based wireless sensor networks: global scale sensing with nano- and pico-satellites. In: Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems (SenSys 2008), pp. 445–446 (2008)
12. Csete, A.: GPredict project. http://gpredict.oz9aec.net/
13. Tan, P.-N., Steinbach, M., Kumar, V.: Association analysis: basic concepts and algorithms. In: Introduction to Data Mining, Chap. 6, pp. 327–413. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (2005)
14. Celandroni, N., et al.: A survey of architectures and scenarios in satellite-based wireless sensor networks: system design aspects. Int. J. Satell. Commun. Network. **31**(1), 1–38 (2013). Wiley
15. Dong, J., Chen, Q., Niu, Z.: Random graph theory based connectivity analysis in wireless sensor networks with Rayleigh fading channels. In: Asia-Pacific Conference Communication, pp. 123–126 (2007)
16. Pottier, B., Lucas, P.-Y.: Dynamic networks NetGen. Technical report, Universit de Bretagne Occidentale, France (2015)
17. Berlin Experimental and Educational Satellite-2 and -3. https://directory.eoportal.org/web/eoportal/satellite-missions/b/beesat-2-3
18. Ting, Y., chunJian, K.: An energy-efficient and fault-tolerant convergecast protocol in wireless sensor networks. Int. J. Distrib. Sens. Netw. 8 pages (2012)
19. Camtepe, S., Yener, B., Yung, M.: Expander graph based key distrbution machinisms in wire sensor networks. In: IEEE International Conference of Communication, pp. 2262–2267 (2006)
20. Muja, M., Lowe, D.G.: Scalable nearest neighbor algorithms for high dimensional data. IEEE Trans. Pattern Anal. Mach. Intell. **36**(11), 2227–2240 (2014)