Computer Science and Information Systems 12(2):895–909 DOI: 10.2298/CSIS141023032S

A Connectivity Monitoring Model of Opportunistic Sensor Network Based on Evolving Graph

Jian Shu[1], Shandong Jiang[1], Qun Liu[1], Linlan Liu[1] and Xiaotian Geng[1]

Internet of Things Technology Institute,
Nanchang Hangkong University,
330063 Nanchang,China
{shujian, liulinlan}@nchu.edu.cn

Abstract. Connectivity is one of the most important parameters in network monitoring. The connectivity model of Opportunistic Sensor Networks (OSN) can hardly be established by traditional graph models due to the fact that its connectivity is timing correlative and evolutionary, which makes it extremely difficult to monitor an OSN. In order to solve the monitoring problem, this paper builds an evolving graph model based on the theory of evolving graph as a description of an OSN. It defines a series of parameters to measure the connectivity of the OSN and establishes an monitoring model. Meanwhile, this paper gives the key algorithms in building the model, the Evolving-Graph-Modeling (EGM) algorithm and the Connected-Journey (CJ) algorithm. The rationality of the monitoring model has been proven by a prototype system and the simulation results. Extensive simulation results show that the proposed connectivity monitoring model can indicate real circumstances of OSN' connectivity, and it is applicable to monitoring an opportunistic sensor network.

Keywords: connectivity monitoring model, opportunistic sensor network, evolving graph.

1. Introduction

In a typical Opportunistic Sensor Network (OSN) [1], mobile sensor nodes move randomly or move according to certain laws to accomplish data gathering. Compared to large-scale static sensor networks, OSN can achieve low-power sensing and large-scale sensing at the same time. It has the same basic features with Opportunistic network (OppNet) [2] and Delay Tolerant Network (DTN) [3], which are intermittent connection, frequently separation and high message delay. An OSN transfers messages according to opportunities created by the movement of nodes, and it is a new type of network whose network status is changing ceaselessly.

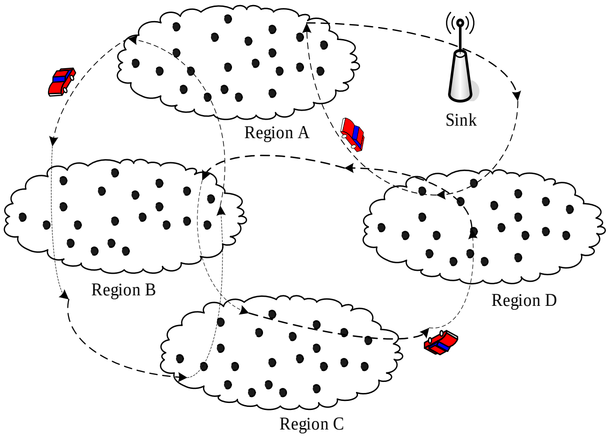

Fig. 1. Opportunistic sensor network

In contemporary OSN, the sensing region is often been cut apart into several separated regions due to the landform and the frequently changing communication quality, as shown in Figure 1. Each region communicates with a mobile node or so called Ferry when it passes by. The connectivity is opportunistic, discontinuous and dynamic caused by changing topology of the network continually.

These features lead to a great challenge to establish a connectivity model of OSN, and to monitor a running one. This paper proposes an evolving graph based monitoring model of OSN. Section 2 analyzes the related researches of connectivity modeling; Section 3 gives the evolving graph based connectivity monitoring model; Section 4 proposes the key algorithms in building the monitoring model and section 5 uses a prototype system and simulation experiments to validate the model.

2. Related Works

Traditional models of network connectivity are basically based on the static graph model, which often focuses on node degree, connectivity of the graph or the probability that a network is k-connected. The topology of OSN is dynamic and timing correlative, for which the study of this kind of network is called a dynamic network model. It mainly focuses on how to describe the dynamic changes in the network topology.

To design a dynamic network model, researchers often use the random graph [4-8] or the random geometric graph [9-11] as a reference model. Paper [12] proposes a random geometric graph by analyzing the features of uniformly distributed network. They find that the nearest neighbor distance of this kind of network conforms to the Poisson distribution. Based on this conclusion they derive probability equations of connected network and k-connected network, and give a function of connectivity probability, which is related to node communication range and the density of the network. Paper [13] presents an analysis of network connectivity of one-dimensional Mobile Ad hoc Networks with a mobility scheme of Random Waypoint Movement. Similar with paper [12], this paper derives a probability function of connectivity by mathematical deduction. Paper [14] proposes a vehicular mobility model in freeway traffic scenarios and analyzes the connectivity parameter of VANET such as the connected probability between any two cars, as well as the diameter size of connected cluster and the number of connected cluster. Such researches focus on dynamic network connectivity, and their purposes are to optimize the design of dynamic sensor networks. However, network monitoring we are studying aims at running networks.

An OSN is notionally and qualitatively similar to an Delay Tolerant Network (DTN). In the preliminary stage of studying these networks, paper [15] proposes the notion of multi-graph, where several edges may exist between pairs of nodes, each weighted with capacity and delay functions. Also, Space-time graph has been defined in paper [16], under perfect information of connectivity at each point in time. Space-time graph is constructed by sampling the instantaneous connectivity over a set of consecutive disjoint time intervals, where topology and link capacity do not change. Such instances are then pasted and linked by directed edges in order to construct the final instance of the space-time graph. Paper [17] provides a notion of an evolving graph, which is a sequence of graphs constructed from the presence schedule of all nodes and links over a set of intervals. Evolving graph can describe the changes and situations of network topology overtime. Paper [18] proves that the problem of building a minimum cost spanning tree in such domain (evolving spanning tree) is NP-hard. Along with the same train of thoughts in paper [17], paper [19] studies the dalian-subgraph problem in evolving graph with the geometric properties between nodes. Mobile-graph is another similar graph model to describe dynamic network. In a mobile-graph, appearance or disappearance of an edge may create a new basic graph, which means that the construction of basic graphs has nothing to do with time interval, but the appearance or disappearance of edges. Therefore, mobile communication can capture the whole process of the evolution of each edge. Paper [20] analyzes network connectivity with mobile-graph. They proposes an approach for protocol assessment based on historical data of mobile-graphs. Paper [22] analyzes the Random Waypoint Movement model, Gauss-Markov Movement model, city area movement model and Manhattan movement model by using the performance evaluation algorithm proposed in paper [19]. Compared to evolving graph, when the evolution of the edges between node pairs is frequent, such as the situation in our work, modeling a mobile-graph is time consuming and it may create a lot of temporary data.

Evolving graph is mainly used in network modeling [23-26] or interplanetary navigation [27]. In summary, evolving graph is able to describe the whole process of the changes of dynamic network topology. It can not only describe dynamic networks themselves, but also can describe their connectivity. In order to monitor the connectivity of running OSN, this paper proposes a connectivity monitoring model based on evolving graph.

3. Modeling and Assumptions

3.1 Scenario and Theoretical Basis

As shown in Figure 1, an opportunistic sensor network is constructed by many sensing nodes in several separated sensing regions, some mobile Ferry nodes and a Sink node. Messages flow form sensing nodes to the Sink node via Ferry nodes when the opportunity arises. In order to describe the dynamic connectivity of OSN, time is usually divided into sequential time slices during the operation of the network. As an example, there are four snapshots taken at different time intervals of a OSN, as shown in Figure 2. where A, B, C and D represent separated sensing regions; E and F are Ferry nodes; S means Sink node.

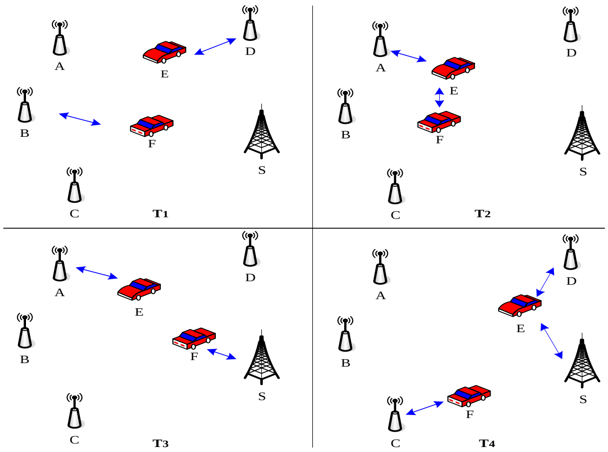

Fig. 2. The evolution of an OSN over time

As one can readily observe, sensing region B and the Sink node S are never connected at a single time interval. Notwithstanding, B can indeed send message to S, using the path over time composed of B,F,S. Surprisingly, this otherwise trivial fact cannot be directly modeled by usual graphs. So now we model it by evolving graph.

Definition 1. Let be given a graph G(V,E) and an ordered sequence of its subgraphs, $S_G = G_1, G_2, \ldots, G_T$ such that $\bigcup_{i=1}^{T} G_i = G$. Let $S_T = t_0, t_1, t_2, \ldots, t_T$ be an ordered sequence of time instance. Then, the system $\Theta = (G, S_G, S_T)$, where each G_i is the subgraph in place during $[t_{i-1}, t_i]$, is called an evolving graph. Let $V_\Theta = V$ and $E_\Theta = E$.

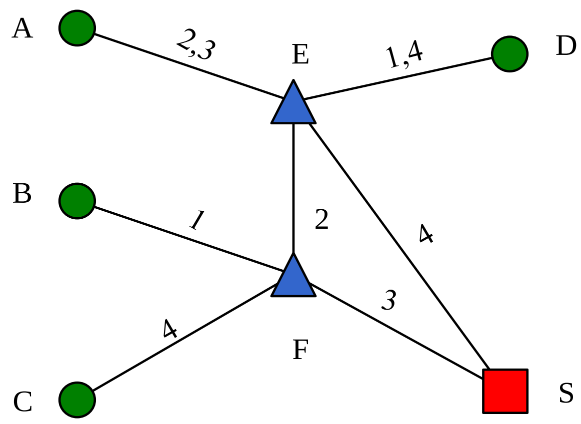

Fig. 3. Evolving graph corresponding to the OSN in Figure 2

Evolving graph represents a formal abstraction of dynamic networks, and can be suited easily to this case. Concisely, an evolving graph is an indexed sequence of T subgraphs of a given graph, where the subgraph at a given index point corresponds to the network connectivity at the time interval indicated by the index number, as shown in Figure 3. Note that B and S is connected within the sequence of B,F,S or B,F,E,S.

Furthermore, in order to monitor an OSN we should transfer the definitions of the most important parameters from usual graph to evolving graph.

3.2 Connectivity Parameters

In static graph models we often use the notion of *path* to represent a connection between a pair of nodes, which means if there exists a path between node A and node B, then we can say A and B is connected. However, if node A and node B are sensing regions in an OSN, the path between A and B is frequently up and down. The notion of path can hardly represent the connectivity in such dynamic networks, so we propose a notion of connected journey to represent the dynamic connectivity in OSN.

Definition 2. Let V_T be a set of unrepeatable vertexes $V_T = v_1, v_2, \ldots, v_k$ with $v_i \in V_\Theta$. Let $S_T = t_0, t_1, t_2, \ldots, t_{k\text{-}1}$ be a time schedule indicating that edge (v_i , v_{i+1}) is to be traversed at time t_i. Then we define a connected journey $CJ_{(v_1, v_k)} = (V_T, S_T)$ if and only if S_T is in a non-decreasing order.

As shown in Figure 3, there are 2 connected journeys between node A and S, which are $CJ_{(A,S)} = ((A, E, F, S), (2,2,3))$ and $CJ'_{(A,S)} = ((A, E, S), (2,4))$. Note that there is no connected journey between C and S because the time schedule is decreasing. It is important to notice that connected journeys connect two nodes over time, even in the case the nodes are never connected in each time slice. Conversely, the fact that two nodes are connected in the underlying graph does not imply the existence of a connected journey between them.

In static graph models, the notion of isolated node and node degree are basically based on the number of neighbors around the node. But in OSN, neighbors are always changing so that the tradition notions cannot describe connectivity accurately. So we give a new notion of node degree based on evolving graph.

Definition 3. If node $V_i \in V_\Theta$ is not a Sink node and the number of connected journeys between V_i and the Sink node V_s is d, we say the node degree of V_i is $D_{V_i} = d$.

Once we have the notion of node degree, we have the notion of isolated node, average node degree, the minimum node degree and k-connected OSN: If nodes have node degree $D_{V_i} = 0$, such nodes are isolated nodes; the average node degree of an OSN $\overline{D} = \frac{1}{N-1}\sum_{i=1}^{N-1} D_{V_i}$; the minimum node degree of an OSN $D_{\min} = MIN(D_{V_1}, D_{V_2}, \ldots, D_{V_{N-1}})$, where N is the number of nodes. If the minimum node degree of an OSN $d_{\min} = k$, we say the OSN is k-connected.

Furthermore, node degree can hardly show the network connectivity directly. For more intuitionistic, we propose the notion of network connectivity degree. Before that, we give successful delivery times of node, which means how many times a node delivers its message to the Sink node successfully.

Definition 4. The successful delivery times of node(S_{V_i}) is the number of such events happen when the node V_i meets a Ferry node in t_i and $t_i \in CJ_{(V_i,S)}$.

For example, as shown in Figure 3, node C meets Ferry F at t_4 for one time, but $CJ_{(C,S)} = NULL$, $t_4 \notin CJ_{(C_i,S)}$, so $S_C = 0$; In another hand, node D meets Ferry E twice in t_1 and t_4, and $t_1 \in CJ_{(D_i,S)}$, $t_4 \in CJ_{(D_i,S)}$, so $S_D = 2$.

Definition 5. If the successful delivery times of node V_i: $S_{V_i} = s_i$, the number of time intervals is Γ, then the node connectivity degree $P_{V_i} = s_i / \Gamma$.

Finally, we propose network connectivity degree to describe the connectivity of an OSN.

Definition 6. Let P_Θ be the network connectivity degree, $P_\Theta = \frac{1}{N-1}\sum_{i=1}^{N-1} P_{V_i}$, where N is the number of nodes in Θ.

As shown in Figure 3, we can obtain that the node connectivity degrees of sensing regions A, B, C and D during 4 time intervals are $P_A = 0.5$, $P_B = 0.25$, $P_C = 0$ and $P_D = 0.5$. According to definition 6, we got the network connectivity degree for that period of time is $P_\Theta = 31.25\%$.

3.3 Connectivity Monitoring Model for OSN

The model we build, as shown in Figure 4, aims at OSN connectivity monitoring. Sensor networks in the bottom layer send messages to a cluster of distributed servers via a Sink node. After classification and extraction, messages will be transferred into a lot of snapshots of the network, which will be represented by a specific data structure (details will be mentioned in section 4); then the servers using a Evolving-Graph-Modeling (EGM) algorithm to translate data into evolving graph model, and a Connected-Journey (CJ) algorithm to get every connected journeys between each sensing region and the Sink node. Finally, the network connectivity degree will be obtained according to definition 4, 5 and 6, which will be sent to the foreground.

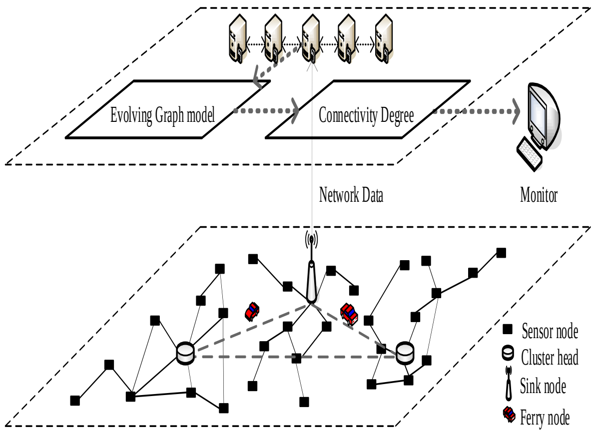

Fig. 4. Connectivity Monitoring Model for OSN

3.4 Assumptions

In order to focus on building the connectivity monitoring, we make the following assumptions:

– We regard the isolated sensing region as a single node, and despite that the regions are constructed by real sensing nodes.
– When capturing network snapshots, we assume that the time intervals are already been divided. Network topology remains constant during each time interval and it is different between contiguous time intervals. In each time interval, messages can be fully exchanged.
– To facilitate the research, we assume that there is only one Sink node in the opportunistic sensor network.

4. Algorithms and Analysis

According to definition 4 to 6, as soon as we transfer network data to a evolving graph model and calculate the connected journey of each sensing nodes, we can obtain the network connectivity degree easily. Now we represent the Evolving-Graph-Modeling algorithm and the Connected-Journey algorithm.

4.1 Evolving-Graph-modeling (EGM) algorithm

According to section 3.1, an evolving graph model is established by network snapshots. Based on the characteristics of OSN, network snapshots can be represented by the neighbor lists of Ferry nodes in each time interval. The neighbor list corresponds to snapshots shown in Figure 2 is shown in Table 1.

Table 1. F period

	Ferry E	**Ferry F**
T1	D	B,C
T2	A,F	E
T3	A	S
T4	D,S	C

An evolving graph can be represented by a special adjacency list. In order to save the time sequences above each edge, we regard different time slices as different items. For example, when we save node A in Figure 3, the physical model is A→(E,1)→(E,2), which means the linked list started with A has two linked nodes: (E,1) and (E,2).

```
Input:   p_tn -- neighbor list of Ferries;
Output: r_al -- adjacency list of the model;
for Each node in p_nt do
  | t_am.add(node,ferry,time_sequence)
end for
for Each time_sequence in t_am do
  | r_al.add(source_node,dest_node,time_sequence)
end for
return r_al
```

Algorithm 1: Evolving-Graph-modeling (EGM)

4.2 EGM algorithm analyzing

Algorithm 1 takes an encounter matrix as an auxiliary space and two independent loops to establish the model. Assume that the number of Ferry is N_f, the number of sensing region is N_a, the number of time slice is N_t. The time and space complexity of Algorithm 1 is described in Theorem 1.

Theorem 1. *The space complexity and time complexity of Algorithm 1 are* $O(N_f * N_a)$ *and* $O(N_f * (N_t + N_a))$.

Proof. The encounter matrix used in Algorithm 1 has a size of N_f * N_a. The construction of the matrix requires traversing the neighbor list, of which the time complexity is N_f * Nt; the construction of the adjacency list requires traversing the encounter matrix, of which the time complexity is N_a * N_f. So the general time complexity is $O(N_f * (N_t + N_a))$, the space complexity is $O(N_f * N_a)$.

Notice that the encounter matrix is not necessary. If we transfer the neighbor list directly to the adjacency list, the time complexity is $O(2*N_f * N_t)$. But practically, $N_t >> N_a$, so that $O(2*N_f * N_t) >> O(N_f * (N_t + N_a))$. Besides, the space of servers are often large enough, so trading space for time is advisable for Algorithm 1.

4.3 Connected-Journey (CJ) algorithm

Finding the connected journeys for an OSM is basically a path searching algorithm. The difference is that, firstly, each time we find a new connectable node, we should check whether the time sequence is in a non-decreasing order; secondly, each node may has many time intervals, so that we should find connected journeys for each time interval individually.

```
Input:   p_al -- adjacency list of the model ; p_sn – the beginning node;
Output:  r_journeyList – connected journey list that been found.
hashTable.init(), stack.init(), journeyTree.init();
add p_sn into journeyTree;
p = p_sn.next;
push p into stack;
while stack is not empty do
    p = stack.top();
    if hashTable[p.node] is Visited or sequence is not increase progressively
        pop stack; break;
    end if
    if p.node is sink node
        add p into journeyTree;
        hashTable[p.node] = visited;
        pop stack;
    end if
    if p.next is empty
        pop stack; break;
    end if
    add p into journeyTree;
    hashTable[p.node] = visited;
    pop stack;
    p = p.next;
    push p into stack;
end while
convert journetTree into r_ journeyList;
return r_ journeyList;
```

Algorithm 2: Connected-Journey (CJ)

Algorithm 2 is based on depth-first search, using a hash table to mark the visited nodes. Before visiting the top element of the stack, we check if it has been visited, and whether the time sequence is in an non-decreasing order (line 8). Each time we found a new appropriate node, we insert the node into a journey tree along with the time instances (line 12, 19). When finishing constructing the journey tree, we transfer the tree into a journey list (line 25).

4.4 CJ algorithm analyzing

Algorithm 2 uses a stack, a hash table and a journey tree as an auxiliary space. We assume that the number of nodes is N, the average number of edges in the connected journey to be found is M, and the average length of the time sequence attached to each node is L. Then the time and space complexity of Algorithm 2 is described in Theorem 2.

Theorem 2. *The space complexity and time complexity of Algorithm 2 are O(N * L) and O(M).*

Proof. Each node in the adjacency list will be visited at most once, and each time instance of node is treated respectively, so the size of the stack is no more than N*L. Likewise, the size of the hash table and the journey tree is no more than N*L. Therefore the space complexity of Algorithm 2 is O(N*L). Based on the number of nodes that already been visited we can reach that the construction time of the journey tree is directly proportional to the length of the time sequence attached to each node. Therefore the time complexity of Algorithm 2 is O(L).

According to Algorithm 1 and 2, we can construct a evolving graph model for a running OSM based on network data and obtain the connected journey of each sensing region. According to definition 4 to 6, the network connectivity degree can be obtained easily.

5. Simulations and Results

5.1 Experiment Parameters

In this paper, we developed a prototype system of the connectivity monitoring model under VS2008, along with the ONE (Opportunistic Network Environment) simulator to simulate an OSN monitoring scenario. Firstly, we simulate an OSN with the ONE simulator, as shown in Figure 5. In this scenario, we extract neighbor lists of Ferries every 10 seconds, which include 120 snapshots of the OSN. Meanwhile, the ONE simulator will count the actual message delivery rate of the OSN. The neighbor lists will be put into the prototype system synchronously. After modeling and processing with the prototype system, the network connectivity degree will be obtained.

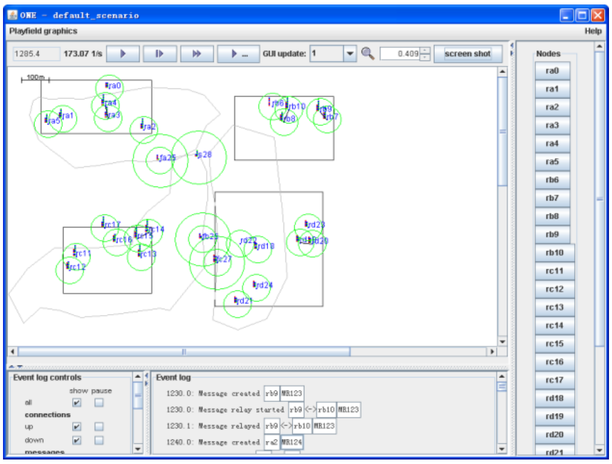

Fig. 5. Experiment scenario in ONE simulator

The total experiment time is 8 hours. During the first two hours, we made all the Ferries working normally; in the second two hours we shut down the Ferry node fa to make the connectivity worse; in the third two hours we shut down fa and fb to make the connectivity even worse; in the last two hours we recover all Ferries. Experiment parameters are shown in Table 2.

Table 2. Experiment Parameters

Parameter	Value
Number of Regions	25
Number of Ferries	3
Sampling period	10 s
Length of time interval	1/12 s
Total sampling time	8 h
Sampled data	Neighbor lists of Ferries

5.2 Results and analysis

The results of network connectivity degree obtained by the prototype system are shown in Figure 6, and the actual message delivery rates counted by ONE simulator are shown in Figure 7.

As shown in Figure 6, during the second experiment period, the network connectivity degree falls down but not very violent, because although the Ferry fa is not working, Ferry fb can also coverage some regions in the top left corner of the scenario, the connectivity loss is not very large. But when we shut down two Ferries fa and fb, the loss of connectivity is significant, as shown in Figure 6 (the green dots). As soon as we recover all Ferries, the connectivity degree is coming back to normal.

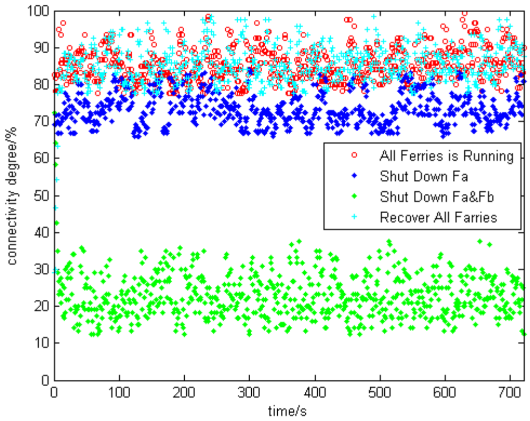

Fig. 6. Simulation results of network connectivity degree

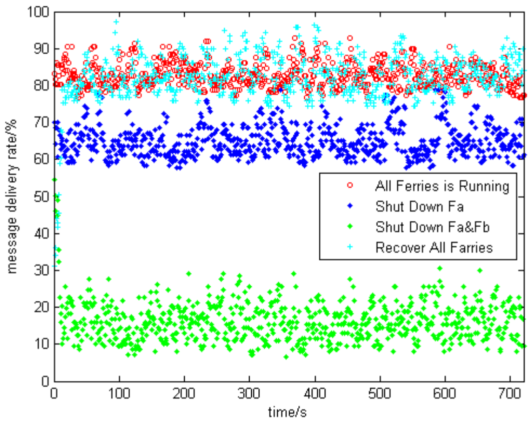

Fig. 7. Actual message delivery rates

Comparing Figure 6 and 7 we can find that the results come from our connectivity monitoring model is consistent with actual network connectivity. The results prove the correctness of the model and also prove that the connectivity monitoring model proposed by this paper is applicable to monitor an opportunistic sensor network.

6. Conclusions

Network connectivity is one of the key concerns in monitoring an OSN. Due to the fact that its connectivity is timing correlative and evolutionary, it can hardly be modeled by usually graph models, which makes it extremely difficult to monitor the connectivity of an OSN. This paper propose a connectivity monitoring model based on evolving graph, by which we can obtain the network connectivity degree of a running OSN synchronously, thereby monitoring the connectivity of the OSN. Extensive simulation results show that the proposed connectivity monitoring model can indicate real circumstances of network connectivity, and it is applicable to monitor an opportunistic sensor network.

Acknowledgments. This paper is supported by the National Natural Science Foundation of China (Grant No. 61363015, No. 61262020), and Project of Production and Research Program of Jiangxi (NO.KJLD14054). Grateful thanks are due to the participants of the survey for their invaluable help in this study.

References

1. Eisenman, S. B., Lane, N. D., & Campbell, A. T.: Techniques for improving opportunistic sensor networking performance. In Proceedings of the 4th IEEE International Conference on Distributed Computing in Sensor Systems. Santorini Island, Greece, 157-175. (2008)
2. Yong-ping, Xiong, Li-min, Sun.: Opportunistic networks. Journal of Software, Vol. 20, No. 1, 124-137. (2009)
3. Zhang, X., Kurose, J., Levine, B. N.: Study of a bus-based disruption-tolerant network: mobility modeling and impact on routing. In Proceedings of the 13th annual ACM international conference on Mobile computing and networking. Multimedia, New York, USA, 195-206. (2007)
4. Simpson, S. L., Moussa, M. N., Laurienti, P. J.: An exponential random graph modeling approach to creating group-based representative whole-brain connectivity networks. NeuroImage, Vol. 60, No. 2, 1117-1126. (2012)
5. Tsitsiashvili, G.: Connectivity probability of random graph generated by point poisson flow. Reliability Theory & Applications, Vol. 8, No. 3, 53-55. (2013)
6. Fronczak, A.: Exponential random graph models. Encyclopedia of Social Network Analysis & Mining, Vol. 36, No. 1, 99-153. (2012)
7. Liu, Liang, Y. Ge, and X. Qiu.: Using exponential random graph (p∗) models to generate social networks in artificial society. In Proceedings of the 2013 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI). Dongguan, Guangdong Province, China, 596-601. (2013)
8. Yagan, O., and A. M. Makowski.: On the connectivity of sensor networks under random pairwise key predistribution. Information Theory, IEEE Transactions on, Vol. 59, No. 9, 5754-5762. (2013)
9. Krishnan, B. Santhana, A. Ganesh, and D. Manjunath.: On connectivity thresholds in superposition of random key graphs on random geometric graphs. In Proceedings of the 2013 IEEE International Symposium on Information Theory Proceedings (ISIT). Istanbul, Turkey, 2389-2393. (2013)
10. Dabirmoghaddam, A., and J. J. Garcia-Luna-Aceves.: Opportunistic walks on Random Geometric Networks and their application in scalability analysis. In Proceedings of the 2013 10th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON). New Orleans, USA, 559-567. (2013)
11. Clementi, A., F. Pasquale, and R. Silvestri.: Opportunistic manets: mobility can make up for low transmission power. IEEE/ACM Transactions on Networking (TON), Vol. 21, No. 2, 610-620. (2013)
12. Bettstetter, C.: On the minimum node degree and connectivity of a wireless multihop network. In Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing. Lausanne, Switzerland, 80-91. (2002)

13. Foh, Chuan Heng, et al.: Network connectivity of one-dimensional MANETs with random waypoint movement. IEEE Communications Letters, Vol. 9, No. 1, 31-33. (2005)
14. Ye, Liu, Guo-xing, Yu.: Study on the connectivity model of VANET based on 802.11p/WAVE and its application. Journal on Communications, Vol. 34, No. 6, 85-91. (2013)
15. Fall, K.: A delay-tolerant network architecture for challenged internets. In Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications. Karlsruhe, Germany, 27-34. (2003)
16. Merugu S, Ammar M H, Zegura E W.:Routing in space and time in networks with predictable mobility. Georgia Institute of Technology, Technical Report, GIT-CC-04-7. (2004)
17. Ferreira A, Jarry A.: Complexity of minimum spanning tree in evolving graphs and the minimum-energy broadcast routing problem. In Proceedings of WiOpt (Vol. 4). University of Cambridge, UK. (2004)
18. Jarry A, Lotker Z.: Connectivity in evolving graph with geometric properties. In Proceedings of the 2004 joint workshop on Foundations of mobile computing. New York, USA, 24-30. (2004)
19. Monteiro J, Goldman A, Ferreira A.: Performance evaluation of dynamic networks using an evolving graph combinatorial model. In Proceedings of IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob'2006). Montreal, Canada, 173-180. (2006)
20. Farago A, Syrotiuk V R.: MERIT: a scalable approach for protocol assessment. Mobile Networks and Applications, Vol. 8, No. 5, 567-577. (2003)
21. de Pebeyre A F, Tarissan F, Sopena J.: On the relevance of the edge-Markovian evolving graph model for real mobile networks. In Proceedings of the IFIP Wireless Days, WD 2013. Valencia, Spain, 1-6. (2013)
22. Von Luxburg, U.: A tutorial on spectral clustering. Statistics and computing, Vol. 17, No. 4, 395-416. (2007)
23. Eiza M H, Ni Q.: An evolving graph-based reliable routing scheme for VANETs. Vehicular Technology, IEEE Transactions on, Vol. 6, No. 24, 1493-1504. (2013)
24. Rossi R A, Gallagher B, Neville J, et al.: Modeling dynamic behavior in large evolving graphs. In Proceedings of the 6th ACM international conference on Web search and data mining. Rome, Italy, 667-676. (2013)
25. Ros F J, Ruiz P M.: Minimum Broadcasting Structure for Optimal Data Dissemination in Vehicular Networks. IEEE Transactions on Vehicular Technology, Vol. 62, No. 8, 3964-3973. (2013)
26. Clementi A, Pasquale F, Silvestri R.: Opportunistic manets: mobility can make up for low transmission power. IEEE/ACM Transactions on Networking (TON), Vol. 21, No. 2, 610-620. (2013)
27. Wanrong W Y L B Y, Baokang Z.: Routing Algorithm for Navigation Constellation Based on Evolving Graph Model. Chinese Space Science and Technology, Vol. 32, No. 5, 76-83. (2012)

Jian Shu is a corresponding author of this paper. He is a full professor at the Nanchang Hangkong University, in China. His current research interests include opportunistic network, machine learning. Contact him at shujian@nchu.edu.cn.

Shandong Jiang is a master candidate at the Nanchang Hangkong University, in China. His research area is opportunistic sensor network. Contact him at 306315953@qq.com.

Qun Liu is a master candidate at the Nanchang Hangkong University, in China. His research area is opportunistic sensor network. Contact him at 304594956@qq.com.

Linlan Liu is a full professor at the Nanchang Hangkong University, in China. Her current research interests include wireless sensor network, embedded system. Contact her at liulinlan@nchu.edu.cn.

Xiaotian Geng is master candidate at the Nanchang Hangkong University, in China. His research area is opportunistic sensor network. Contact him at hellasmoon@yahoo.com.

Received: October 23, 2014; Accepted: May 15, 2015.