Research on Community-Based Opportunistic Network Routing Protocol

Zhanwei Liu(✉), Wenbo Yuan, Na Su, and Hui Wang

School of Electronics and Automation, Inner Mongolia Electronic Information Vocational Technical College, Hohhot 010070, Inner Mongolia, China
lzw981@qq.com

Abstract. The opportunity network is realized through the internode movement to achieve inter-network communication. There is not a complete communication path between the source node and the destination node in the network. This paper studies the opportunity network routing protocol based on the community. First, comparing the advantages and disadvantages of GN and K-means two community partition algorithm, Select the algorithm with high accuracy, high data transmission rate, and delay of the small GN algorithm. Then the GN algorithm is applied to the and Spray and Wait routing protocol. Regardless of the size of the network, the node density, The protocol is scalable and can maintain good performance.

Keywords: Opportunity network · GN · K - means · Spray and Wait routing protocol

1 Introduction

In recent years, wireless communication technology has developed at an unprecedented speed, but traditional multi-hop wireless networks, such as Wireless Sensor Network [1], MANET [2], and Wireless Mesh Network [3], do not interrupt the wireless network connection and split Cellular mobile communication network technology has strong network coverage, but it is obviously insufficient in data service support; wireless local area network has developed rapidly in recent years, but its coverage is very limited, thus promoting Opportunistic Networks [4,5] the birth and development. Opportunistic networks are a new type of network that can transmit information when the wireless link is down and there is no end-to-end path. It is regarded as an important direction for the development of mobile AdHoc networks, which is of great significance to ubiquitous computing.

Early routing algorithms in opportunistic networks include flood-based Epidemic Routing [6], direct transmission routing algorithms, Spray and Wait [7], Spray and Focus [8], and Prophet [9] algorithms, but the bandwidth occupied by Epidemic Routing Larger, it will waste network resources, and the direct transmission algorithm is a single copy routing algorithm. Although the network

W. Li and D. Tang (Eds.): MOBILWARE 2020, LNICST 331, pp. 31–43, 2020.
https://doi.org/10.1007/978-3-030-62205-3_3

overhead is small, the transmission success rate is low and the average delay is large. The Spray and Wait routing algorithm include two phases: spray and wait. In the spraying phase, each time it connects with other nodes, it will allocate 1/2 message copy to the other party. After the number of message copies is 1, it enters the waiting phase. In the waiting phase, the node holds the message until it meets the destination node. Spray and Focus routing algorithm is an improved algorithm of Spray and Wait, which changes the waiting phase into the focusing phase. In the focusing phase, the node selects the appropriate forwarding node based on the utility value. The Prophet algorithm [10] combines two types of forwarding methods, infection forwarding, and encounter prediction. Based on the forwarding probability of the meeting node to the destination node, it is determined whether to forward messages to the meeting node. Therefore, it can limit the number of copies in the network and improve the transmission success rate.

Peng et al. [11] proposed a quota routing algorithm DPER based on delivery probability prediction. This algorithm predicts the delivery probability of other nodes based on the local information of the node and determines the allocated message copy quota based on the delivery probability. Pan Hui et al. [12] proposed a routing algorithm Bubble Rap based on social attributes. This routing algorithm judges the forwarding node based on the destination community and the ranking obtained from the node centrality. When the message is outside the destination community, the forwarding node is judged according to the global ranking; when the message is inside the destination community, it is judged according to the local ranking.

Community division is to divide the nodes in the network into multiple communities according to the strength of the relationship between the nodes. The community division algorithms in opportunistic networks are mostly based on the community division algorithms of complex networks. The more classic algorithms include the GN algorithm [13] and the K-means algorithm [14] and so on.

2 Comparative Analysis of Community Division Algorithms

2.1 Community Division Algorithm

GN Algorithm

The GN algorithm was proposed by Girvan and Newman in 2002, and it is a representative splitting method. The edge intermediary is defined as the number of shortest paths through any two nodes of the edge in the network. Continuously deleting the edge with the largest intermediary from the network will divide the entire network into more closely related communities.

In the community structure detection algorithm, the denseness of the reasonable division of the internal connection of the community should be higher than the expected level of the random connection network, and the Q function is used to quantitatively describe the modularity of the community division. Suppose

the complex network has been divided into n communities, then first define an $n \times n$ dimensional symmetric matrix e, where element e_{ij} represents the proportion of the edges connecting nodes in community i and community j to all sides, and $Tre = \sum_i e_{ii}$ of this matrix represent all the ratio of the edges that connect the nodes in the community to the total number of edges [15][16][17], define the total value $a_i = \sum_i e_{ij}$ of the column (or row) to represent the ratio of all edges connected to nodes in community i to the total number. According to the definition of e_{ij} and a_i, the Q function can be expressed as:

$$Q = \sum_i (e_{ii} - a_i^2) = Tre - \|e^2\|$$

Where $''\|e^2\|''$ is the modulo of the matrix, that is, the sum of the elements in e^2.

If the number of edges between nodes within the community is not as many as those obtained by random connection, the value of the Q function is negative. When the value of the Q function approaches 1, it indicates that the division of the community structure is reasonable. In practical applications, the value of Q is generally between 0.3 and 0.7.

The basic flow of the GN algorithm is as follows:

(1) Calculate the edge median of each edge in the network;

(2) Find the edge with the largest number of edges and remove it from the network;

(3) Recalculate the edge intermediaries of the remaining edges in the network;

(4) Repeat steps (2) and (3) until each node is a separate community;

Take the data source of karate club as an example, run the GN algorithm, and get the results shown in Fig. 1:

```
The original network was:
1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34

Program analysis shows that the original network is likely to split into 2 communities:
community1: 3  9  10  15  16  19  21  23  24  25  26  27  28  29  30  31  32  33  34
community2: 1  2  4  5  6  7  8  11  12  13  14  17  18  20  22

The key edges leading to the split and the process of splitting are:

Based on the original networkRemove the edge (edge with the largest intermediary):
v(1, 32), v(1, 3), v(1, 9), v(14, 34), v(20, 34), v(3, 33), v(2, 31), v(2, 3), v(3, 4), v(3, 8), v(3, 14),
Will be split into 2 groups, the situation after the split is:
community1: 3  9  10  15  16  19  21  23  24  25  26  27  28  29  30  31  32  33  34
community2: 1  2  4  5  6  7  8  11  12  13  14  17  18  20  22
Process finished with exit code 0
```

Fig. 1. Results from running in the Eclipse environment

Divided results into two communities:

Society 1 {3, 9, 10, 15, 16, 19, 21, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34}

Society 2 {1, 2, 4, 5, 6, 7, 8, 11, 11, 12, 13, 14, 17, 18, 20, 22}

Draw the results of community division with NetDraw software, as shown in Fig. 2:

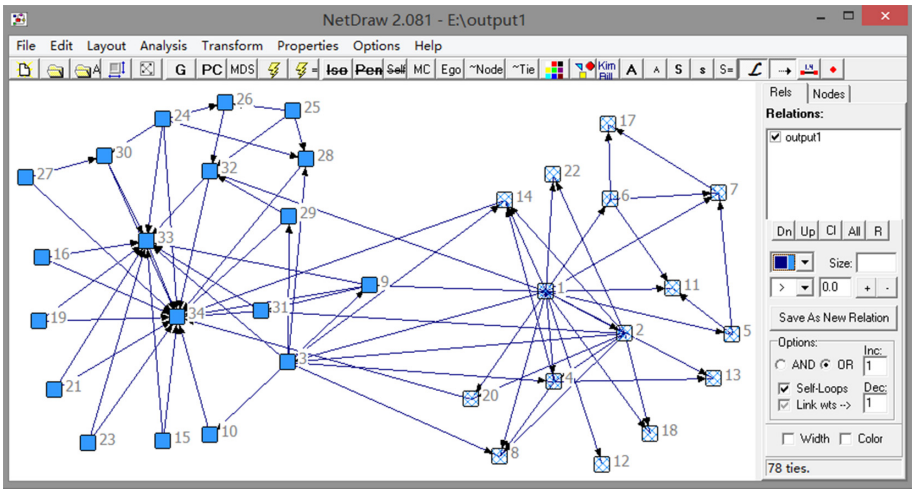
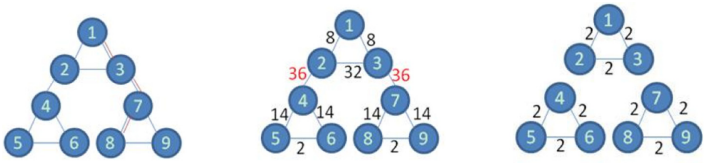


Fig. 2. The results of community division drawn by NetDraw software

Take Fig. 3 as an example to illustrate the execution flow of the GN algorithm:

1. Use the shortest path algorithm to find the shortest path from vertex 1 to vertex 8 in Fig. 3(a) (red part in the figure)
2. Repeat step 1 to detect the shortest path between all vertices of the network, and calculate the edge intermediaries of all edges, as shown in Fig. 3(b).
3. Count the maximum number of edges and delete it to get the community structure shown in Fig. 3(c).
4. Run the GN algorithm to get the results shown in Fig. 4:



(a) Shortest Path (b) Edge Intermediate (c) Remove the edge with the largest edge intermediary

Fig. 3. Execution process of GN algorithm (Color figure online)

```
"C:\Program Files\Java\jdk-12.0.1\bin\java.exe" "-javaagent:G:\IntelliJ IDEA\install\IntelliJ I
The original network was:
1  2  3  4  5  6  7  8  9
Program analysis shows that the original network is likely to split into 3 communities
community1: 4  5  6
community2: 7  8  9
community3: 1  2  3
The key edges leading to the split and the process of splitting are:

Based on the original networkremove the edge (edge with the largest intermediary): v(2,4),
Will be split into 2 groups, the situation after the split is:
community1: 4  5  6
community2: 1  2  3  7  8  9

and remove the edge (edge with the largest intermediary): v(3,7),
Will be split into 3 groups, the situation after the split is:
community1: 4  5  6
community2: 7  8  9
community3: 1  2  3
Process finished with exit code 0
```

Fig. 4. Running results of GN algorithm

From the experimental results, the first example uses a karate club data source. The result of community division using the GN algorithm is exactly the same as the result of community division. The second example uses an ordinary graph as the data source and GN The community division results obtained by the algorithm are the same as the actual ones, so the community division accuracy of the GN algorithm is high.

K-Means Algorithm

K-means algorithm is a typical algorithm in common clustering algorithms. First, the value of the parameter k needs to be determined, and then the set of n objects is divided into k clusters by the distance of the distance, so that the clustering result is that the distance within the cluster is close. The similarity is high, and the distance between clusters is long, the similarity is low. The similarity of a cluster is a measure of the mean of the objects in the cluster and can be regarded as the center of mass or center of gravity of the cluster.

1. Three points of K-means algorithm for clustering

(1) Select a certain distance as the similarity measure between data samples

When using the K-means clustering algorithm, the Euclidean distance is often used to calculate the distance between data samples. Besides, you can also choose Manhattan distance as the similarity measure of the algorithm according to actual needs. That is to say, given a data set $X = \{Xml, m = 1, 2, \cdots, total\}$, the samples in X use d description attributes $A_1, A_2, \cdots, A_d$, and d attribute descriptions

are all continuous attributes. Given data samples $X_i = (X_{i1}, X_{i2}, \cdots, X_{id})$ and $X_j = (X_{jl}, X_{j2}, \cdots, X_{jd})$, where $X_{i1}, X_{i2}, \cdots, X_{id}$ and $X_{jl}, X_{j2}, \cdots, X_{jd}$ are the similarities between samples X_i and X_j respectively, usually expressed by the distance $d(X_i, X_j)$ between them. The smaller the distance is, the more similar the samples X_i and X_j are, and the smaller the difference is; the larger the distance is, the less similar the samples X_i and X_j are, and the greater the difference is. Euclidean distance is calculated by Eq. (1):

$$d(X_i, X_j) = \sqrt{\sum_{k=1}^{d} (X_{ik}, X_{jk})^2} \tag{1}$$

(2) Choose a criterion function to evaluate clustering performance

The squared error and criterion function are methods used by the K-means clustering algorithm to evaluate clustering performance. Detailed description: Given a data set X, it is assumed that X contains k clustering subsets $X_1, X_2, \cdots, X_k$; it only includes descriptive attributes and does not include category attributes. The number of samples in each subset is $n_1, n_2, \cdots, n_k$; the average representative points (also called cluster centers) of each clustering subset is $m_l, m_2, \cdots, m_k$.

Then the squared error is expressed as Eq. (2):

$$E = \sum_{i=1}^{k} \sum_{p \in X_i} \|p - m_i\| \tag{2}$$

2. K-means algorithm description

(1) Selection method of initialization cluster center.

a. Rely on the experience to judge and analyze the data samples, and then select C suitable initial cluster centers from the sample set;

b. Select the 1st-C sample data of the entire sample set as the initial cluster center;

c. The entire sample set is then group C, and the average value of all sample data in each group is calculated, which is the C initial cluster centers;

(2) The initial clustering is based on the distance between the sample data and each cluster center, and the samples are assigned to the nearest class. The specific process is to first take a sample and assign it to the class represented by its nearest cluster center according to the nearest principle, and then classify all samples into the appropriate class according to this method and recalculate the cluster. Class center, that is, the sample mean, update the cluster center, and perform iterative operations.

(3) The error square is used to judge whether the clustering is reasonable, and the classification is modified if it is not reasonable. Judge and modify repeatedly until the algorithm termination condition is reached.

3. K-means algorithm example

Suppose the data object set S is shown in Table 1. As a two-dimensional sample for cluster analysis, the required number of clusters is $k = 2$.

Table 1. Data object collection

Point	X coordinate	Y coordinate
O_1	2	0
O_2	3	2
O_3	3	0
O_4	0	6
O_5	1	7

(1) Select $O_1(2,0)$, $O_2(3,2)$ as the initial cluster center, that is, $M_1 = O_1 = (2,0)$, $M_2 = O_2 = (3,2)$.

(2) For the remaining data objects, the data is assigned to the nearest cluster according to the distance between the data and the center of each cluster.

To O_3 :

$d(M_1, O_3) = \sqrt{(3-2)^2 + (0-0)^2} = 1$

$d(M_2, O_3) = \sqrt{(3-3)^2 + (0-2)^2} = 2$

$d(M_1, O_3) < d(M_2, O_3), Coming\ from\ O_3\ to\ C_1$

In the same way we can get $d(M_2, O_4) < d(M_1, O_4)$, that is, O_4 is assigned to C_2; $d(M_2, O_5) < d(M_1, O_5)$, that is, O_5 is assigned to C_2, and new clusters $C_1 = \{O_1, O_3\}$, and $C_2 = \{O_2, O_4, O_5\}$ are obtained after the update.

Calculate the square error criterion with a single variance of:

$E_1 = [(3-2)^2 + (0-0)^2] = 1$

$E_2 = [(0-3)^2 + (6-2)^2] + [(1-3)^2 + (7-2)^2] = 49$

The overall mean-variance is: $E = E_1 + E_2 = 50$.

(3) Calculate the center of a new cluster.

$M_l = ((2+3)/2, (0+0)/2) = (2.5, 0)$

$M_2 = ((3+0+1)/3, (2+6+7)/3) = (1.33, 5)$

(4) Repeat 2, 3 to get O_1, then O_2, O_3 assigned to C_l, O_4 and O_5 assigned to C_2.

(5) Update to get new clusters $C_1 = \{O_1, O_2, O_3\}$ and $C_2 = \{O_4, O_5\}$. The center is $M_1(2.67, 0.67)$ and $M_2(0.5, 6.5)$, the single variance is $E_1 = 3.33$ and $E_2 = 1$, and the overall average error is $E = 4.33$.

It can be seen from the above that after the first iteration, the overall average error is reduced from 50 to 4.33, and the cluster center is unchanged, so the iteration process is stopped and the algorithm ends.

4. K-means algorithm execution steps

a. Determine k initial cluster centers;

b. Assign the remaining nodes to the nearest cluster according to the minimum distance principle;

c. Calculate the mean of each cluster and use it as the new cluster center;

d. Repeat steps b and c until the cluster center no longer changes;

e. The end, get k clusters;

5. The experimental process uses a set of 3D data sources and community division results, as shown in Table 2:

Table 2. Data object collection and community division results

Point	X coordinate	Y coordinate	Z coordinate	Affiliate
O_1	100	10	20	0
O_2	15	50	30	0
O_3	74	88	20	0
O_4	91	66	88	0
O_5	36	88	36	0
O_6	23	13	10	1
O_7	22	17	18	2
O_8	56	57	55	1
O_9	52	59	100	1
O_{10}	80	78	60	3
O_{11}	73	19	20	1
O_{12}	53	28	10	1
O_{13}	65	72	30	3
O_{14}	67	31	50	1
O_{15}	48	92	90	4
O_{16}	0	28	10	1
O_{17}	74	95	70	4
O_{18}	16	73	10	2
O_{19}	85	15	20	3
O_{20}	62	0	50	1
O_{21}	58	36	50	1
O_{22}	19	8	20	1
O_{23}	59	45	80	1
O_{24}	25	52	30	1
O_{25}	45	48	50	1
O_{26}	46	57	30	1
O_{27}	22	54	22	1
O_{28}	88	34	50	3
O_{29}	53	77	90	4
O_{30}	11	71	40	2
O_{31}	30	56	40	1
O_{32}	0	51	8	1
O_{33}	24	63	40	1
O_{34}	92	32	90	3
O_{35}	87	83	30	3
O_{36}	46	26	40	1
O_{37}	98	93	100	3
O_{38}	34	71	20	4
O_{39}	94	12	20	3
O_{40}	33	38	30	1
O_{41}	26	28	44	1
O_{42}	90	53	40	3
O_{43}	79	39	20	3
O_{44}	59	26	19	1
O_{45}	55	52	48	1
O_{46}	10	91	33	2
O_{47}	21	59	77	1
O_{48}	57	62	22	3
O_{49}	68	16	55	1

According to Table 2, the data division results shown in Fig. 5 can be obtained:

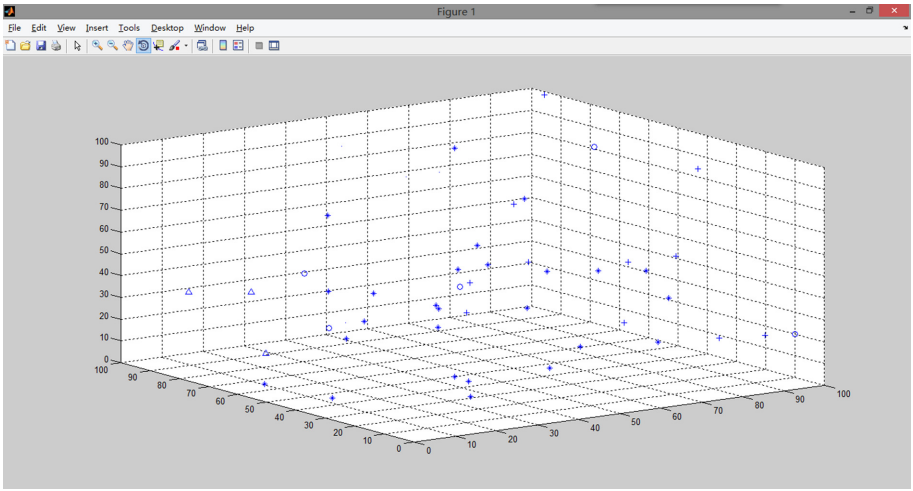


Fig. 5. K-means division of 3D data

The processing results are shown in Fig. 6:

According to Fig. 6, we can see that the K-means algorithm is less accurate than the GN algorithm, so the GN algorithm is selected.

2.2 Comparative Analysis of GN Algorithm and K-Means Algorithm

Analysis of GN Algorithm

The community division effect of the GN algorithm is more accurate than other algorithms, and it is suitable for medium-scale networks with the number of nodes n less than 10,000.

The disadvantage of the GN algorithm is that the time complexity is relatively large, so the analysis effect on large-scale complex networks is not ideal, and it takes a long time. The time complexity is $O(n^3)$, and n is the number of nodes. To solve this problem, people have proposed many improved algorithms based on the GN algorithm, such as a Newman Fast Algorithm based on the GN algorithm, which actually is a condensed algorithm based on the idea of greedy algorithm, which can analyze a complex network of 1 million nodes.

Analysis of K-Means Algorithm

K-means algorithm is simple and fast. It can efficiently process large-scale data sets and is suitable for continuous data sets. When the result cluster is dense, the cluster is distinct from the cluster, and the clustering is better when the data distribution is convex or spherical.

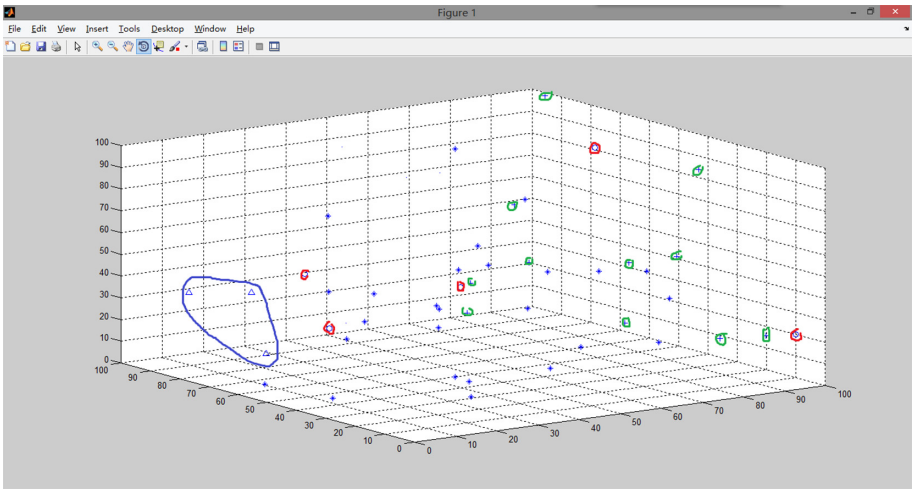


Fig. 6. Processing of K-means results

K-means clustering is an iterative clustering algorithm. During the iterative process, the members in the cluster are continuously moved until the ideal cluster set is obtained. The time complexity is $O(t \times k \times n)$ and t is the number of iterations. The final clustering result of the K-means clustering algorithm largely depends on the initial center point, which is randomly selected. The traditional K-means algorithm may converge to a local optimum instead of a global optimum. Some methods to improve K-means performance have been proposed, but most of them require additional input, such as a threshold for the number of data points in the data set. The performance of the K-means clustering algorithm depends on the initial position of the clustering center, that is, the random initial value selection may lead to different clustering results, and even there are no solutions, and the algorithm performs "noise points" and "solitary points" The data is very sensitive, a small amount of this type of data will have a very large impact on the average.

Solution: Set different initial values and compare the calculation results until the result stabilizes, but this is time consuming and wastes resources.

Although the K-means algorithm is simple and fast, the experimental results are too dependent on the choice of initial values, so K-means is not used as a routing protocol. Although the time complexity of the GN algorithm is relatively large, the accuracy is relatively high. For the accuracy of the experiment, the GN algorithm is selected as the routing protocol.

2.3 Community-Based Routing Protocols

Spray and Wait (SW) algorithm [18] is a routing algorithm based on the community in this experiment. It is a multi-copy routing scheme proposed by Spyropoulos et al. The SW algorithm is divided into two phases. In the S phase,

each message will generate a corresponding copy and sent to multiple different forwarding points, so that some data packets in the source node will be diffused to neighboring nodes; in the W phase, if there is no When the destination node is found, multiple nodes holding the copy of the message switch to the direct sending state and forward the message to the destination node.

The Spray and Wait algorithm has two modes, Binary mode, and non-Binary mode. In Binary mode, the algorithm mechanism is that when the source node encounters a new relay node, it sends half of the data packets to the new relay node, leaving half of the data packets on its own; the source node and the relay node repeat the above process until When there is only one data packet in all nodes, the node transfers to the Wait phase and adopts direct transmission to the destination node. The Spray and Wait algorithm has an L parameter, which describes the number of packets.

The Spray and Wait algorithm is based on a flooding strategy, combining the fastness of infectious routing, the simplicity, and simplicity of direct transmission, and is committed to effectively balancing delay and energy consumption. The transmission rate is high, the transmission delay is small, and it is close to the optimal; It has good scalability and can maintain good performance regardless of the size of the network and the change in node density [18,19].

3 Experimental Analysis of Spray and Wait Routing Protocol

As shown in Fig. 7, there are five communities, namely:

$C1 = \{S, 1, 2, 3\}$, $C2 = \{4, 5\}$, $C3 = \{12, 13\}$, $C4 = \{7, 8, 10, 11\}$, $C5 = \{D, 14, 15, 16\}$

The process of sending a message from the source node S in the community C1 to the destination node D in the community C5 is as follows:

(1) If node S and node D are in the same community, the message is directly forwarded in the community; otherwise, step (2) is performed;

(2) Node S and Node D are not in the same community, and then send the message to the neighboring communities C3 and C4;

(3) If the destination node D has been encountered, perform the forwarding in step (1), otherwise, continue to forward;

(4) Until the destination node D receives the message;

During the entire process of message forwarding above, during the Spray phase, a copy of each message in the community will be generated and sent to the nodes of the neighboring community, so that the data packets in the source node will be diffused to the surrounding communities. The destination node is not found during the message forwarding process, and the forwarding is continued until the destination node receives the data.

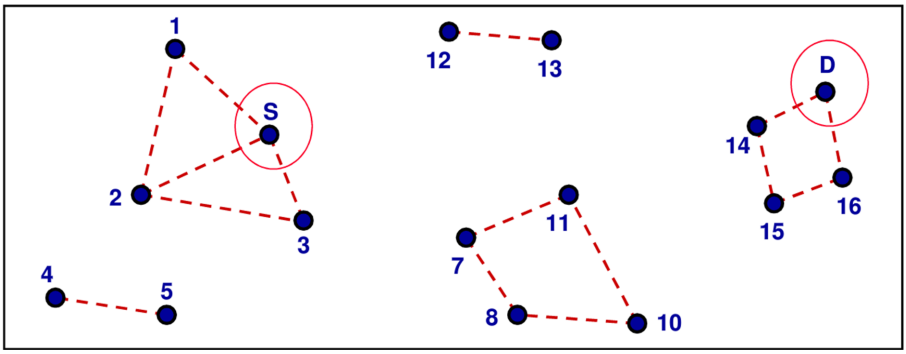


Fig. 7. Spray and wait routing

4 Conclusion

This paper proposes a community-based opportunistic network routing protocol. The research is mainly divided into two parts. The first part is based on the existing opportunistic network routing protocol to propose the community-based opportunistic network routing protocol. First, compare the advantages and disadvantages of the two community division algorithms of GN and K-means, decide to use a more accurate GN algorithm for community division, and then perform message routing on the divided communities. This paper uses the Spray and Wait routing protocol, which is fast, simple, and has a high transmission rate.

Acknowledgment. The authors wish to thank Inner Mongolia Higher Education Research Project under Grant NJZY17472.

References

1. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: A survey on sensor networks. IEEE Commun. Mag. **40**(8), 102–114 (2002)
2. Loo, J., Mauri, J., Ortiz, J.: Mobile ad hoc networks. Current Status and Future Trends, Mobile Ad Hoc Networks (2012)
3. Akyildiz, I.F., Wang, X.: A survey on wireless mesh networks. IEEE Commun. Mag. **43**(9), S23–S30 (2005)
4. Stavroulaki, V., et al.: Opportunistic networks. IEEE Veh. Technol. Mag. **6**(3), 52–59 (2006)
5. Yong-Ping, X., Li-Min, S., Jian-Wei, N., Yan, L., et al.: Opportunistic networks. J. Softw. **20**(1), 124–137 (2009)
6. Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: Energy-efficient communication protocol for wireless microsensor networks. In: Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, p. 10. IEEE (2000)
7. Vahdat, A., Becker, D., et al.: Epidemic routing for partially connected ad hoc networks (2000)

8. Spyropoulos, T., Psounis, K., Raghavendra, C.S.: Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In: Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-tolerant Networking, pp. 252–259 (2005)
9. Spyropoulos, T., Psounis, K., Raghavendra, C.S.: Spray and focus: efficient mobility-assisted routing for heterogeneous and correlated mobility. In: Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerComW 2007), pp. 79–85. IEEE (2007)
10. Lindgren, A., Doria, A., Lindblom, J., Ek, M.: Networking in the land of northern lights: two years of experiences from dtn system deployments. In: Proceedings of the 2008 ACM Workshop on Wireless Networks and Systems for Developing Regions, pp. 1–8 (2008)
11. Min, P., Pei-Lin, H., Kai-Ping, X., Han-Cheng, L.: Delivery probability prediction based efficient routing in DTN. Chin. J. Comput. **34**(01), 176–183 (2011)
12. Hui, P., Crowcroft, J., Yoneki, E.: Bubble rap: social-based forwarding in delay-tolerant networks. IEEE Trans. Mob. Comput. **10**(11), 1576–1589 (2010)
13. Newman, M.E.: Assortative mixing in networks. Phys. Rev. Lett. **89**(20), 208701 (2002)
14. Chou, X.: The Improvement and Application of K-Means Algorithm. Ph.D. thesis, Beijing University of Posts and Telecommunications (2012)
15. Yu, Z., Yu-dong, L., Zhao, J.: Vector similarity measurement method. Tech. Acoustics **28**(4), 532–536 (2009)
16. Sun, J.G., Liu, J., Zhao, L.Y.: Clustering algorithms research. J. Softw. **19**(1), 48–61 (2008)
17. Fu-ding, X., Da-wei, Z., et al.: Density set algorithm search for communities in complex networks. J. Univ. Electr. Sci. Technol. China **40**(4), 483–490 (2011)
18. Peleg, S., Friedlander, B.: The discrete polynomial-phase transform. IEEE Trans. Signal Process. **43**(8), 1901–1914 (1995)
19. Xianda, Z., Bao: Non-Stationary Signal Analysis and Processing. National Defense Industry Press, Beijing (1998)