InterestSpread: An Efficient Method for Content Transmission in Mobile Social Networks

Ning Wang
Dept. of Computer and Information Sciences
Temple University
Philadelphia PA 19122
ning.wang@temple.edu

Jie Wu
Dept. of Computer and Information Sciences
Temple University
Philadelphia PA 19122
jiewu@temple.edu

ABSTRACT

In Mobile Social Networks (MSNs), the single-path routing might not have enough of a chance to transmit content to the destination (i.e., low network throughput), due to limited contact opportunities. Meanwhile, the multiple-path routing improves the network throughput at the cost of higher system resource consumption (e.g., energy and storage). Therefore, there exists a trade-off between the network throughput and the system resource consumption. Moreover, we should consider user features in MSNs, i.e., some of the nodes would like to help the other nodes with the same social features (e.g., neighbors, classmates) during content transmission, regardless of their resource consumption. These nodes are called *interested nodes*. The remaining nodes, called *uninterested nodes*, will be reluctant to transmit contents to save their resources. To achieve high network throughput and control the system resource consumption of uninterested nodes, we propose a novel multiple-path two-stage routing algorithm, *InterestSpread*, to transmit contents in the MSNs as follows. (1) In the first stage, we limit the content transmission into a relay candidate set. The contact information, bandwidth information, and social features are leveraged together to select such a set. (2) In the second stage, a classical max-flow method is used to get maximum throughput in the relay candidate set. The simulation based on real human and synthetic traces indicate that our algorithm achieves a good trade-off between throughput and the system recourse consumption.

Categories and Subject Descriptors

C.2.2 [**Network Protocols**]: Routing protocols; G.2.2 [**Graph Theory**]: Network problems, Graph algorithms

General Terms

Algorithm, Design, Theory

MSCC'14, August 11, 2014, Philadelphia, PA, USA.

http://dx.doi.org/10.1145/2633675.2633677.

Keywords

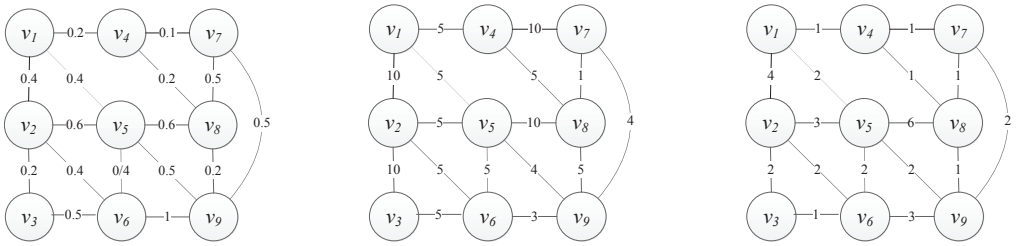
Connected dominating set (CDS), mobile social network (MSNs), max-flow, relay candidate set.

1. INTRODUCTION

The evolution of mobile devices and wireless communication in the recent years has made mobile social networks (MSNs) attract more attentions. MSNs are used for content transmission such as sharing music and movies in a delay-tolerant network (DTN) environment: The property of DTN is that there might not exist a contemporaneous end-to-end path. In this way, contents are buffered for extended intervals of time until an appropriate forwarding opportunity is recognized in hopes that it will eventually reach its destination (i.e., store-carry-forward). Many routing algorithms [8, 13, 6, 2, 14, 1, 10, 7, 9, 3] based on the store-carry-forward paradigm have been proposed to solve routing problems in such a scenario. Basically, the majority of the existing algorithms focus on how to get better routing performance (e.g., high delivery ratio and network throughput and low overhead ratio and delay) and try to control the consumption of system resource (e.g., energy and storage) at the same time.

According to [4], an enormous amount of content will be generated in MSNs, but only a very small portion of users will be interested in receiving any of it. Users in MSNs are self-publishing consumers, which means that constant waves of new videos and the convenience of the web are quickly personalizing the viewing experience, leading to a great variability in user behavior and attention span. This kind of User Generated Content (UGC) [4] network is dynamic and decentralized, which has reshaped traditional content dissemination. A challenging question thus arises as to how to distribute relevant content to interested users without disturbing uninterested users too much. In this paper, we consider a practical network model, in which nodes are divided into two types according to the content sources: One type is that of *interested nodes*, which would like to help the source to transmit content due to the same social features. And the second are *uninterested nodes*, which are reluctant to transmit contents to save their resources.

In this paper, we present InterestSpread, a novel algorithm for content dissemination in MSNs to balance the routing performance and system resources. It is a trade-off between the routing performance (network throughput) and system resource (consumption of uninterested nodes). The idea is that we want to control the system resource consumption of uninterested nodes and achieve *relatively high* network throughput at the same time. The content is preferably



(a) Contact graph (b) Bandwidth graph (c) Relative bandwidth graph

Figure 1: Build the relative bandwidth graph

transmitted among interested nodes to reach the destination. However, if uninterested nodes can greatly improve the network throughput, the uninterested nodes will work as relays to improve the network throughput.

This paper makes the following contributions.

- We explore three aspects of network information together, i.e., contact, social, and bandwidth information, for a more practical routing decision. It achievses high performance with little system resouce.
- A novel two-stage routing algorithm, InterestSpread, is proposed. In our method, a relay candidate set is selected. limiting the relay selection into the relay candidate set achieve a good trade-off between performance and system resource usage.

The remainder of the paper is organized as follows. We introduce the related work about content dissemination in mobile social networks in Section II. The network model is introduced in Section III. The InterestSpread algorithm is presented in Section IV. The evaluation setting and the simulation results are shown in Section V. After that, we conclude the paper in Section VI.

2. RELATED WORK

The basic assumption in MSNs is that people with similar social features have a higher probability of meeting with each other than do strangers. There exists two main advantages by using social feature information: (1) the social features is stable for a relatively long time, and thus, prediction based on social features is more reliable. (2) the social feature information has a smaller maintenance cost than does the contact history information. Some famous examples are Bubble [8], PeopleRank [11], etc. In Habit [10], the physical information and the social information are combined in a dominated way to transmit content. Though it can minimize the nodes which are not interested in the content to act as relays, the delay can be considerably long.

However, few of algorithms combine the system constraints and social features together. We argue that these approaches are not suitable for exchanging large amounts of content in MSNs. On the one hand, the size of content is increasing at an amazing speed, especially along with the high-solution videos and photos' wide usage; A single contact opportunity might not provide enough contact opportunity due to the bandwidth constraint. On the other hand, a large amount of contents are always being generated in MSNs. Popular nodes should not act as relays every time, otherwise they will quickly drain their limited resources. How to leverage the network throughput and system usage is a question.

3. NETWORK MODEL

In MSNs, due to the limit contact opportunity, the content might be divided into many small packets through each contact opportunity. Thus, several routes might be used together to transmit the content. An example of the application scenario is that a person wants to share a piece of music, or a short video, with his friend in the university within 2 hours. Due to bandwidth constraints and unpredictable contact duration, the content is splitted into smaller packets, and then be forwarded through several contact opportunities. In the reminder of this section, we provide the details of the network model. We do not consider the buffer constraint of the network. This is reasonable, since the storage is really cheap and mobile devices have high storage compared to the size of the content.

3.1 Contact Information Estimation

Through neighbor exchange informaion, nodes gradually have a view of the network topology. The contact probability of p_{ij} is computed in the following way. Node i checks its contact record with node j in the time interval $[0, T]$. Then, node i sums every contact duration, t_{ij}^k in this time interval and compares it with the whole time interval, as follows.

$$p_{ij} = \frac{\sum_k t_{ij}^k}{T} \tag{1}$$

3.2 Relative Bandwidth Transformation

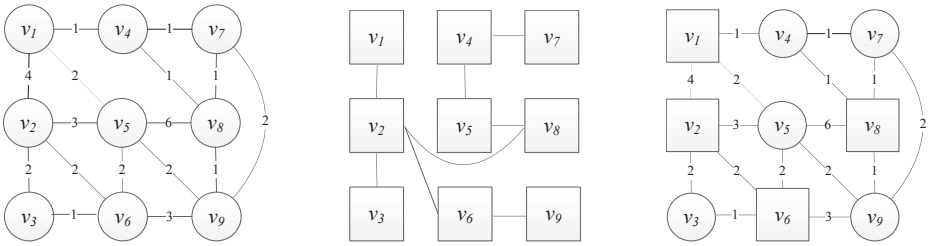
The bandwidth information of each node is transmitted in the same way as the contact information. After a certain period of time, each node has a view of the bandwidth information of the network, as shown in Figure 1(b). Based on contact opportunity estimation, we modify the bandwidth information to *relative bandwidth information*. We can roughly combine contact probability and bandwidth by using the following equation.

$$B_{ij} = \alpha \times \bar{B}_{ij} \times p_{ij} \tag{2}$$

where α is an empirical value that we can get from experiments, and $\bar{B}_{ij}$ is the actual bandwidth between node i and node j. The idea is to determine how much content transmitted during each contact opportunity. In Figure 1(c), the capacity of content transmitted from (v_1, v_2) should be four times larger than that of (v_1, v_4).

3.3 Transmission Graph Transformation

In this paper, interest can be regarded as nodes who have same social features (e.g., colleagues, classmates). Intuitive-



(a) Relative bandwidth graph (b) Interest graph (c) Transmission graph

Figure 2: Build the transmission graph

ly, the node with the same features would like to help each other. The interest information can be added into the relative bandwidth graph when we make routing decisions. If we want to transmit contents from source node v_2 to destination node v_8, this process can be implemented in Figure 2(c). The nodes interested in receiving content from node v_2 are marked (square nodes) in the relative bandwidth graph, and thus, form the transmission graph for content transmission from v_2 to v_8. The transmission graph combines three kinds of information, and later we will use the transmission graph to make a routing decision.

4. ALGORITHM

In this section, we propose InterestSpread algorithm, which limits the content transmission within a relay candidate set to balance the demand for network throughput and system resource consumption, using uninterested nodes as relays.

4.1 Finding a Connected Dominating Set

To get a relay candidate set, we find a connected dominating set (CDS) firstly. A CDS of a graph G is a set S of vertices with two properties: (1) Any node in S can reach any other node in G by at least one path that stays entirely within S. That is, S is a connected subgraph of G. (2) Every vertex in G either belongs to S or a neighbor to a node in S. For example, in Figure 2(c), nodes v_2, v_5, v_8 form a CDS. This is because all the other nodes, $v_1, v_3, v_4, v_6, v_7, v_9$ are the neighbor of nodes v_2, v_5, v_8. By using a CDS, we can guarantee getting a connected subgraph of the whole graph. Besides, there is at least one path from source to destination in the network if a CDS exists.

In this paper, we use a distributed algorithm proposed in [5] to find a CDS. We first assign priorities to nodes, according to the node's interest and the topology information. We can regard the priority of a node as the node's id in [5]. Intuitively, the interested node should have higher priority than the uninterested node. A node with a better topology (e.g., higher node degree, wider bandwidth, etc.) should also have higher priority. We calculate node i's sum of bandwidth, $B_i = \sum_j B_{ij}$, to estimate the maximum amount of content node i can transmit. The priority setting rule is that interested nodes will always have higher priority than uninterested nodes. For a pair of interested or uninterested nodes, the node which has larger B_i will have higher priority.

Marking Principle: All the nodes are unmarked initially, then through nodes' neighbor set information exchange, the nodes which exist two unconnected neighbors are marked and these marked nodes form a CDS.

From the marking process of Figure 3(b), the marked nodes (dark color) form a CDS, V'. Though we can get a CDS from the marking principle, the size of CDS formed by marking principle is relatively huge, and thus, we further propose a pruning principle prune the CDS by Rule k.

Pruning Principle: We denote $N(i)$ to represent the neighbor set of vertex i, and $N(V'_k)$ to represent the neighbor set of a vertex set V'_k, that is, $N(V'_k) = \bigcup_{i \in V'_k} N(i)$. After pruning by Rule k, we get a CDS, V''.

Rule k. Assume that $V'_k = \{v_1, v_2, \ldots, v_k\}$ is the vertex set of a connected subgraph in G'. If $N(i) - V'_k \subseteq N_R(V'_k)$ in G and $id(i) < min\{id(1), id(2), \ldots, id(k)\}$, then we can remove i from CDS.

Take the example of Figure 3(c), nodes, v_7, v_8, v_9 and their neighbor sets are covered by v_8, $\{v_6, v_8\}$, and $\{v_1, v_2, v_6, v_8\}$ respectively. So we can prune v_7, v_8, v_9 from the selected connected dominating set. We can prove that Rule k will not destroy the property of CDS.

Theorem 1. *If V' is a connected dominating set of a undirected graph G, and V'_R is the set of vertices removable under Rule k, then set V'', which equals to $V' - V'_R$ is also a connected dominating set of G.*

Proof. It is clear that $|V'| = 1$ is right. This is because $V'' = V'$. If $|V'| > 1$, for every vertex i in G, it is either in $V^{'}$ or not in V'. If $i \notin V'$, it is dominated by at least one vertex in V' due to the propriety of CDS. If $i \in V'$, it is also dominated by a vertex in V', since V' is connected. In addition, there always exists a vertex $j \in V'$ satisfying $id(j) = max\{id(w)\colon\ w \in N(i)\}$, which cannot be removed by applying Rule k. Therefore, i is dominated by at least one vertex $j \in V''$. Assume that the graph made up by V' is not connected. If we put back the removed vertices one by one in reverse order of pruning V', we can find node i, which reconnects V''; that is, after the removal of i, at least one pair of vertices loses its connecting path. However, this is impossible. If i is removed from V'' by applying Rule k, its neighbor set is covered by vertices with higher id's than $id(i)$. So there always exists another path between the nodes in node i's neighbor set. Therefore, removal of i cannot make V'' unconnected, which is a contradiction. □

4.2 Adjusting the Relay Candidate Set

We can further adjust the relay candidate set to improve the network throughput or delete the uninterested nodes, which cannot bring enough expected throughput. It is easy to understand that we should always add the node which can get the high throughput, and delete the node which contributes little network throughput. However, it is hard (or needs much computation) to get the exact throughput

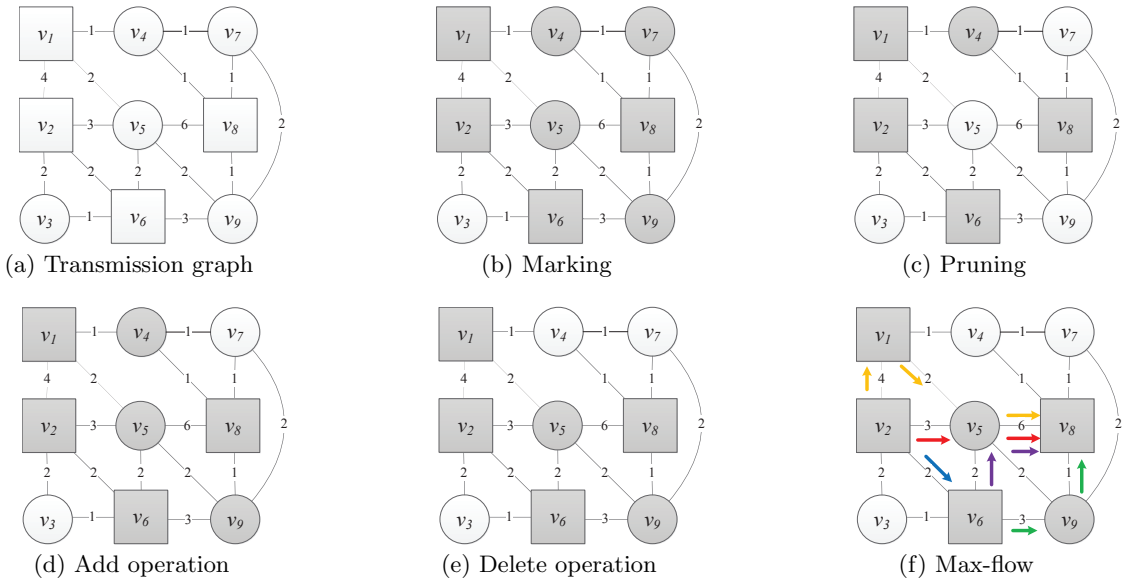


Figure 3: The routing process of InterestSpread

contribution of a node. Intuitively, nodes with high degree and wide bandwidth are possible to provide large throughput. In this paper, we use the expected throughput, E_i, to estimate the throughput. We calculate E_i by using the following equation.

$$E_i = \sum_{j \in N(V'')} B_{ij} \tag{3}$$

Add Operation: All the interested nodes which do not belong to the relay candidate set are added first. The idea is that interested nodes would like to forward the contents, and will have the potential to increase the throughput of the network without extra cost. After that, we would like to add uninterested nodes with good topologies. The uninterested nodes are sorted according to their expected throughput. We add the node which has the largest expected throughput iteratively. After each iteration, we should re-calculate the expected throughput. Also, we should set a threshold, β, for *add operation*, and should stop if we find that the node with largest expected throughput is not that high (smaller than β). β is an empirical value, and we can adjust it according to the demand. In Figure 3(d), there are four nodes, v_3, v_5, v_7, and v_9 which do not belong to the relay candidate set. First, we calculate the expected throughput for the white nodes v_3, v_5, v_7, and v_9. $E_5 = 13$, $E_9 = 4$, $E_3 = 3$, and $E_7 = 2$. If $\beta = 5$, we add node E_5 first. After we add v_5, we calculate E_9, E_7, and E_3 again. $E_9 = 6$ $E_7 = 4$, and $E_3 = 3$. We keep doing this until the node with largest expected throughput $E_7 = 4$, which is smaller than β.

Delete Operation: Similar to the *add operation*. We calculate the expected throughput of each node which belongs to the relay candidate set. If there exist uninterested nodes in the relay candidate set which can only provide limited expected throughput, we will delete them iteratively. For example, In Figure 3(d), the relay candidate set consists of $v_1, v_2, v_4, v_5, v_6, v_8, v_9$, while the expected throughput of each node is 7, 9, 2, 15, 7, 8, 6, respectively. We should delete E_4 from the relay candidate set, since E_4 is smaller than the threshold, $\beta = 5$. However, we need to avoid deleting the articulation point, where nodes can cause the relay candidate set to become unconnected, should they be deleted. After delating node v_4, we calculate the expected throughput of nodes $v_1, v_2, v_5, v_6, v_8, v_9$ again. The expected throughput is 6, 9, 15, 7, 8, 6, respectively. Then, we find that the expected throughput of any remaining node is larger than the threshold. The *delete operation* finishes, and we get the relay candidate set R.

4.3 Calculating the Maximal Throughput

After the relay candidate set is selected, the system resource consumption of uninterested nodes is controlled. Then, how to get the maximum network throughput in the selected relay candidate set becomes our concern. Max-flow algorithm is a good solution for this. In optimization theory, the max-flow problem involves finding a feasible flow through a single-source, single-sink flow network that is the maximum. It should meet two constraints. First, the capacity constraints, the bandwidth constraints, cannot exceed its maximum value, as shown in equation 4. Second, the sum of the contents entering a node must equal the sum of the contents exiting a node, except for the source and the sink nodes, as shown in equation 5.

$$\forall ij \in E, \quad f_{ij} \leq B_{ij} \tag{4}$$

$$\sum_{ij \in E} f_{ij} = \sum_{ij \in E} f_{ji} \tag{5}$$

where E is the edge of graph G and f_{ij} is the content from node i to node j. If we find a path from the source to the destination, we adjust the bandwidth of edges along the path until we cannot find such a path. Then, we get the maximum network throughput. In Figure 3(f), we can calculate the maximum network throughput from the relay candidate set $v_1, v_2, v_5, v_6, v_8, v_9$. The network throughput is 6. If we consider all the nodes from the network to be the relay candidate, the network throughput is 7. This example shows

Algorithm 1 InterestSpread

Input: Transmission graph G
Output: Network throughput
1: Mark Transmission graph G and get a CDS, V', according to the marking principle.
2: Assign each node an id, according to its topology and interest information
3: Prune V' and get V'', according to the pruning principle.
4: Conduct add operation and delete operation to V'' and get relay candidate set R.
5: Maxflow in relay candidate set R.

that InterestSpread can reduce the usage of the system resource while keeping a high throughput.

5. EVALUATION

In this section, we report the performance evaluation of our algorithm, based on the real trace *Infocom2006* trace [12], which has been widely used in MSNs routing simulation. Besides, we also do simulations on synthetic datasets to analyze our algorithm in a general environment.

5.1 Simulation Settings

In *Infocom2006* trace, groups of participants are asked to carry small devices (iMotes) for four days during the INFOCOM 2006 conference. The contact information of the 78 participants are recorded in the iMotes. In addition, each participant was asked to fill a questionnaire with a number of questions about themselves (e.g., the information about their nationalities and the interested topics). Clearly, we use the answers of participants to estimate their social information. In the simulation, we use nationalities, languages, and interested topics as interest information to form interested and uninterested nodes. Since all the participants use the same devices, it is reasonable to assign the same bandwidth to each participant. Then we get the contact, interest, and bandwidth information of the network. In synthetic datasets, we randomly set the contact, bandwidth, and interest information for 100 simulation rounds to generate a general simulation environment.

5.2 Algorithms in Comparison

The goal of InterestSpread algorithm is to leverage high throughput and system resource consumption. In order to quantify the extent to which InterestSpread achieves this goal, we will evaluate InterestSpread in the network throughput and the usage of uninterested relay nodes together. Basically, InterestSpread is compared with three kinds algorithms, which are different in selecting relay sets. They are *Rand*, *ContactOnly* and *InterestOnly*. Rand algorithm chooses the relay candidate set randomly without considering any information. ContactOnly selects the relay nodes according to nodes' relative bandwidth in the same way as we have mentioned in the above section. The InterestOnly limits the relay candidate nodes into the interested nodes, and then performs selection. The four algorithms contact add operation and delete operation in the same way. We change the threshold β from 0 to 100 to evaluate these four algorithms. $\beta = 0$ means all the nodes are selected in the relay candidate set. Along with β increases, we delete more nodes in the relay candidate set. The network throughput and system resource consumption decrease at the same time. When $\beta = 100$, the network throughput and relay candidate set is relatively stable so we stop here.

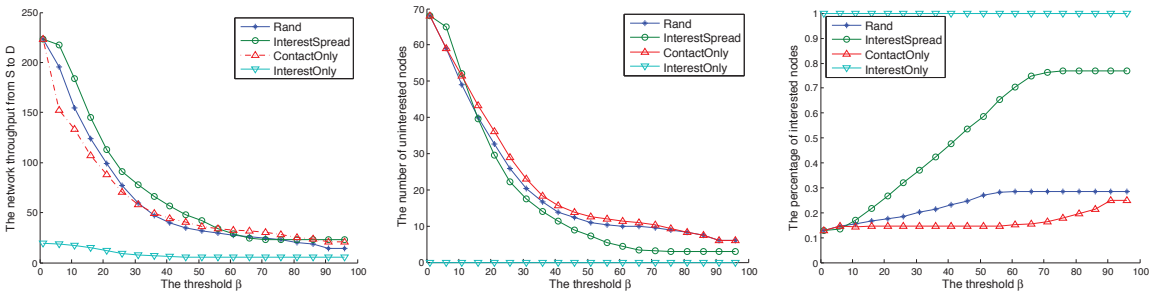
5.3 Evaluation Results

In *Infocom2006*, the result of Figure 4(a) indicates that the throughput of all the four algorithms decrease along with the increasing of β. Among these four algorithms, InterestSpread and ContactOnly achieve the top performances. When it comes to the comparison between InterestSpread and ContactOnly, the InterestSpread achieves better network throughput, due to the help of interested nodes. When β continues to increase, the ContactOnly algorithm achieves a little higher network throughput since only the nodes with best topologies are still remaining after extensive delete operations. However, if we consider the number of uninterested nodes in the relay sets that the two algorithms, we find that the ContactOnly uses more uninterested nodes than that does InterestSpread when β is increasing, which is not what we expected. In Figure 4(c), the percentage of interested nodes in the relay candidate set is increasing in InterestSpread. However, the ContactOnly has the lowest percentage while using interested nodes. From Figure 4, we can conclude that InterestSpread can achieve better or the same, network throughput with less system resource consumption. As for InterestOnly, InterestOnly consumes the least amount of system resource and it is no wonder that it achieves a relatively low network throughput.

We generate extensive different settings in contact, bandwidth, and interest information in synthetic dataset. We get similar conclusions as those of *Infocom2006*, but they are much more clear, as shown in Figures 5(a) and 5(b). The InterestSpread algorithm can achieve better throughput while using fewer uninterested nodes. In Figure 5(c), the percentage of interest nodes in InterestSpread is much higher than in Rand and ContactOnly, except when all the relay nodes are deleted and the source transmits content to the destination directly. The simulation results from the synthetic datasets show that InterestSpread can achieve about 20% and near 40% more than ContactOnly, *Rand* respectively in network throughput, but use 40% fewer uninterest nodes when β is not big. If threshold β continues to increase, the difference between these three algorithms decreases, since most nodes are deleted from the relay candidate set.

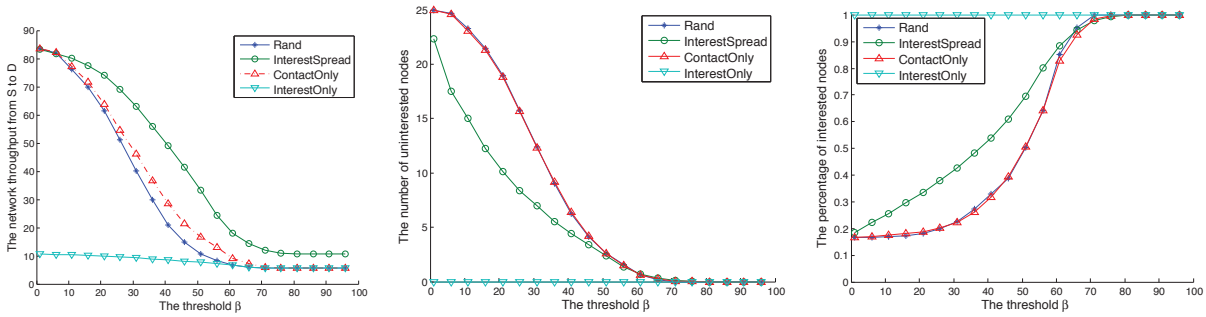
6. CONCLUSION

In this paper, we study a special type of mobile social network, where the system resource consumption problem is considered. Nodes would like to help other nodes with the same feature but are reluctant to help nodes without the same feature. A novel two-stage routing algorithm, InterestSpread, is proposed. InterestSpread leverages contact information, social relationship, and bandwidth constraint together to make a routing decision. First, a relay candidate set is selected to control system resource consumption while achieving high expected network throughput at the first time. Then, a classical max-flow method is used to achieve the maximum network throughput in a selected relay candidate set. Extensive simulation shows that our algorithm can balance the network throughput and system resource consumption. Our future work will focus on the study of a similarity of nodes, where nodes' different features are con-



(a) Network throughput (b) Number of uninterested nodes (c) Percentage of interested nodes

Figure 4: Performance comparisons of InterestSpread vs other three algorithms in *Infocom2006*



(a) Network throughput (b) Number of uninterested nodes (c) Percentage of interested nodes

Figure 5: Performance comparisons of InterestSpread vs other three algorithms in synthetic datasets

sidered together; besides, multi-source and multi-destination content transmission with the system resource consumption problem will be further studied.

7. REFERENCES

[1] A. Balasubramanian, B. Levine, and A. Venkataramani. Dtn routing as a resource allocation problem. In *Proceedings of the ACM SIGCOMM*, pages 373–384, 2007.

[2] C. Becker and G. Schiele. New mechanisms for routing in ad hoc networks through world models. In *Proceedings of the CyberNet Plenary Workshop*, page 2, 2001.

[3] Y. Cao and Z. Sun. Routing in delay/disruption tolerant networks: A taxonomy, survey and challenges. *Communications Surveys*, 15(2):654–677, 2013.

[4] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon. I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system. In *Proceedings of the ACM SIGCOMM*, pages 1–14, 2007.

[5] F. Dai and J. Wu. An extended localized algorithm for connected dominating set formation in ad hoc wireless networks. *Parallel and Distributed Systems*, 15(10):908–920, 2004.

[6] Y. Dai, P. Yang, G. Chen, and J. Wu. Cfp: Integration of fountain codes and optimal probabilistic forwarding in dtns. In *Proceedings of the IEEE GLOBECOM*, pages 1–5, 2010.

[7] X. F. Guo and M. C. Chan. Plankton: An efficient dtn routing algorithm. In *Proceedings of the IEEE SECON*, pages 550–558, 2013.

[8] P. Hui, J. Crowcroft, and E. Yoneki. Bubble rap: Social-based forwarding in delay-tolerant networks. *Mobile Computing*, 10(11):1576–1589, 2011.

[9] S. Jain, K. Fall, and R. Patra. Routing in a delay tolerant network. In *Proceedings of the ACM SIGCOMM*, volume 34, pages 145–158, 2004.

[10] A. J. Mashhadi, S. Ben Mokhtar, and L. Capra. Habit: Leveraging human mobility and social network for efficient content dissemination in delay tolerant networks. In *Proceedings of the IEEE WoWMoM*, pages 1–6, 2009.

[11] A. Mtibaa, M. May, C. Diot, and M. Ammar. Peoplerank: social opportunistic forwarding. In *Proceedings of the IEEE INFOCOM*, pages 1–5, 2010.

[12] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau. CRAWDAD data set cambridge/haggle (v. 2009-05-29). http://crawdad.cs.dartmouth.edu/cambridge/haggle.

[13] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *Proceedings of the ACM SIGCOMM*, pages 252–259, 2005.

[14] H. Zheng, Y. Wang, and J. Wu. Optimizing multi-copy two-hop routing in mobile social networks. In *Proceedings of the IEEE SECON*, 2014.