International Journal of Science and Engineering
http://140.133.4.61/IJSE/index.html

Using Stationary Relay Nodes (Thrown Boxes) to Maximize Message Forwarding Performance in Delay-Tolerant Networks

Nguyen Huu Minh Anh and Chih-Lin Hu

Department of Communication Engineering

National Central University

101523601@cc.ncu.edu.tw; clhu@ce.ncu.edu.tw

Abstract

This paper addresses the efficiency on message forwarding over heterogeneous nodes in DTNs. Among the approaches for routing protocols in DTNs, the strategy of deploying the stationary relay nodes called “thrown boxes” is considered for increasing network performance efficiently. Till now, few studies have provided routing solutions for thrown-box-based network paradigms in the DTN research field. The technical content of this paper includes three parts. Firstly, this study considers the deployment of thrown boxes that can assist in message delivery in delay-tolerant networks. A utility-based algorithm is applied to make the message forwarding decision between two nodes whenever they are in contact. Secondly, we study where to appropriately deploy thrown boxes under various scenarios. Finally, this study runs simulations to evaluate the proposed scheme in terms of message delivery probability. Performance results show that the proposed algorithm achieves high performances and thrown boxes play an important role in message forwarding performance.

Keywords: *Thrown boxes, utility-based algorithm, delay-tolerant networks.*

1. INTRODUCTION

In mobile ad hoc networks (MANET) [1] environments, mobile nodes can directly connect with neighboring nodes to distribute and exchange data that they carry during movements without the need of wireless infrastructure support. Popular ad hoc routing protocols, such as ad-hoc on-demand distance vector (AODV) and dynamic source routing (DSR), assume that there exist a complete end-to-end path from any source and destination nodes. However, such an assumption maybe vulnerable in delay-tolerant networks (DTNs) [2] that lack instantaneous end-to-end paths. In DTNs, it is not judicious to take classical store-and-forward routing protocols that are well understood in MANET. The study of DTNs must address the problems of intermittent connection, long transfer delay, and low message delivery ratio in such environments. Alternatively, this design adopts the store-carry-and-forward approach, as shown in Fig. 1, where data are incrementally moved and stored throughout the network in hopes and will eventually reach their destinations.

Previous research works proposed several routing algorithms in DTNs. Specifically, the replica-based technique attempts to maximize the advantage of unscheduled forwarding opportunities to increase the delivery ratio and reduce delivery time. This is feasible only when the networks can provide large amounts of local storage and internode bandwidth relative to expected message traffic and resource waste. The utility-based technique (e.g. SimBet [11]) uses the social metrics to estimate or predict the potential relay nodes may meet the destination. However, due to the uncertainty of future encounters and underlying social graph, it is quite possible that this technique fails to deliver the message to the destination successfully.

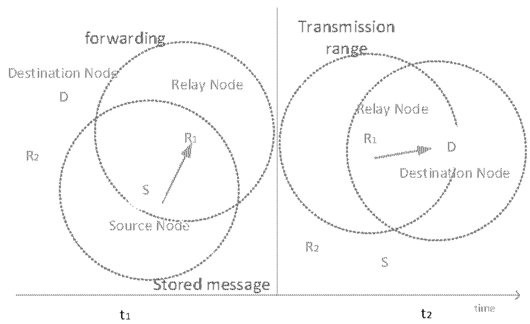

Figure 1: The store-carry-forward approach

Among the approaches for routing protocols in DTNs, the strategy of deploying the stationary relay nodes called “thrown boxes” is considered for increasing message delivery probability and reducing the message overhead

*Corresponding author: clhu@ce.ncu.edu.tw

DOI : 10.6159/IJSE.2014.(4-4).06

efficiently. Thrown boxes are firstly introduced by Zhao and Chen [3]. Thrown boxes are small and inexpensive devices equipped with wireless interfaces and storage. When two nodes pass by the same location at different times, the thrown box acts as a relay, creating a contact opportunity where none existed before. Fig. 2 shows an example of using thrown boxes in a mobile DTN. At time t_1, node S sends data to thrown boxes. At a later time t_2 when node R moves close to the thrown boxes, it receives S's data from the thrown boxes.

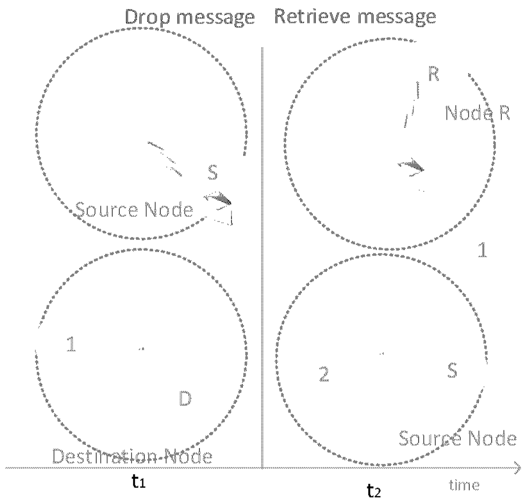

Figure 2: Thrown boxes deployment

However, the network topology evolves over time due to node mobility. This address more challenges into the messages forwarding over thrown boxes assisted Delay-tolerant networks. Moreover, the need to find an appropriate places for thrown boxes deployment to maximize network performance.

The main contributions of this paper are:

1) We propose the utility-based algorithm exploiting the advantage of thrown boxes in DTNs.

2) We study how to deploy thrown boxes under various scenarios so that the performance of routing algorithm can be improved efficiently.

2. RELATED WORKS

The routing protocols used in [5] [6] are to improve the delivery message probability being transferred to the destination by replicating the copies of messages. On the other hand, epidemic routing protocol mainly relies on flooding-based scheme without history information. Specifically, whenever mobile nodes come into contact with other nodes, they disseminate their stored messages to any neighboring nodes that have not received such message before. Though this scheme can improve the delivery ratio, it involves considerable resource expenses and high overhead.

As the ProPHET protocol in [9], this exploits the advantage of history tables to compute the delivery predictability of the nodes. Each node maintain the delivery predictability of every other nodes. When two nodes come into contact, they exchange their history tables. Thus, this information is utilized to forward the messages to the one that has higher frequency to reach the destinations. In [10], Maxprop protocol maintains a complicated history collection from all encountered nodes. This information is built through a cost of any possible routing path. The messages are forwarded to other nodes in specific order that takes into account the cost and message deliver probabilities based on previous encounters. Further, the RAPID protocol [4] designs different utility measures used to manipulate the message queue to minimize missed deadlines, average delay and maximum delay.

The study presented in [7] [8] is another significant work in routing algorithms for DTNs. In that study, authors proposed the algorithm which can perform fewer transmissions than flooding based routing schemes and higher delivery ratios. There are two different routing schemes included: Spray and Wait [7] and Spray and Focus [8]. In Spray and Wait, the source tries to spread a limited number of copies of a message over the network to relay nodes. Then wait until a relay node meets the destination. On the other hand, Spray and focus is designed to eliminate some deficiencies of Spray and Wait. According to a using appropriate utility-function, it can forward a message closer to its destination and achieve a better performance than Spray and Wait. However, these methods create a lot of redundant messages that could rapidly increase the overhead in resource-scare DTNs.

In addition, a lot of research also uses some auxiliary nodes to relay messages. Research in [12] exploits thrown boxes to relay message, and research in [13] uses mobile "message ferries" to relay transfer messages, etc. Other studies [14] [15] [16] [17] on thrown boxes mainly focus on the capacity and delivery delay of the Epidemic algorithm when putting thrown boxes into Delay-tolerant networks. In [3], the authors study the joint thrown box deployment and routing optimization problem. However, their focus is only on the long term average capacity.

3. SYSTEM MODEL

In our network scenarios, we consider the DTNs which is composed of mobile nodes and thrown boxes. Each node visits high frequently to thrown boxes areas, while other locations are visited less frequently. For example, we consider students in a university campus; the thrown boxes areas are the cafeteria, classrooms, and sport areas. Specifically, let $N = (1, 2, 3, ..., n)$ be the set of all mobile nodes in DTNs. A node is identified by a unique ID and node mobility is assumed to be driven. Let $B = (B_1, B_2, ..., B_b)$ be the set of thrown boxes in a network. When one node moves to another node's (or thrown box's) transmission coverage, they can exchange the copies in their buffers.

In general, the thrown boxes are deployed at selected locations, to store a copy of message and also create opportunities for other nodes to retrieve those copies and carry them to other locations. Therefore, the copy is stored at thrown box for sustained periods. The relay pattern of using thrown boxes as follows: when mobile node visits thrown boxes area at a given time, it drops the copies of message to an encountered thrown box. Later, the destination node can retrieve message by searching in visiting thrown box before this message is expired. Suppose that there are no transmissions between any pair of thrown boxes.

In our model, we take the advantage of thrown boxes to increase the message forwarding performance in DTNs. The copies could be carried by many mobile nodes throughout the network before reaching the thrown boxes. A node has different visiting probability to thrown boxes due to its mobility. A notation $\lambda(B_i^j)$ denotes the visiting probability that mobile node j visits thrown box B_i. In addition, we assume all mobile nodes move independently and the visiting process is based on discrete time. It is clear that the visiting probability will influence the distributions of the copies at thrown boxes and also the message forwarding performance in DTNs.

Symbol	Description
n	Number of mobile nodes
b	Number of thrown boxes
$\lambda(B_i^j)$	The probability that node j visits thrown box B_i
$U(j)$	The utility function of node j

4. THE UTILITY-BASED THROWN BOXES ALGORITHMS (U-Thrown boxes)

The objective of this section is to present more details the utility-based thrown boxes algorithm for message forwarding in DTNs. To come up with our objective, we introduce the utility function and U-Thrown box algorithm with a given number of message copies. In order to improve message forwarding performance, we also propose various schemes to deploy thrown boxes under different scenarios.

4.1 The Utility Function

We first study the utility function. The objective of this function is to make the forwarding decision whenever two nodes come into contact. Specifically, when node i holding a copies encounters with node j, the utility function is exploited to decide whether node i hands over the copies to node j or not. According to assumptions about our network model, the destination node has higher probability of visiting thrown boxes than that of meeting another node. Thus, the message forwarding performance reaches to good performance when thrown boxes hold a significant number of copies, then the destination can retrieve the copies when it reaches to thrown box later. Our detailed utility function methods are presented as follows.

$$U(j) = \frac{\sum_{j \in N}^{i \in B} \lambda(B_i^j)}{b} \tag{1}$$

We consider the visiting probability of mobile nodes at thrown boxes. Thus, we denote the probability that thrown box b receives any copy from node j at a given time by (1). This also means that the higher number of mobile nodes (with high visiting probability) having copies is the better performance of network. In other words, the visiting frequency plays an important role in determine which node is "better" for message forwarding process.

4.2 The U-Thrown boxes phases

The Utility-based thrown boxes algorithm has three phases: spraying, spreading and retrieving.

1) In the spraying phase, the source sends

copies quickly to others encountered nodes. The spraying method using in this algorithm is similar to Binary Spay&Wait [7]. More precisely, the source of a message initially starts with L copies; any node A that has $n > 1$ message copies (even source or relay), and encounters another node B (with no copies), hand over to B $\lfloor n/2 \rfloor$ and keeps $\lceil n/2 \rceil$ for itself. Moreover, if a node visits thrown boxes, it drops all messages to thrown boxes. When every node is left with less than one copy ($n \leq 1$), it switches to forwarding phase.

2) In the forwarding phase, every node is holding with less than one message copy left ($n \leq 1$).The utility function U is performed to make a forwarding decision. When a node i (having a message copy) encounters with another node *j*, we compare the utility between two nodes. If $U(j) > U(i)$, then node i will forward its message copy to node *j*, else node *i* still keeps it. Moreover if a node *i* visits thrown box, if node *i* has more than one token, it will hand over half of its tokens. If not, node *i* forward the copy to thrown box when node *i* has only one token.

3) The retrieving phase is performed in parallel with the forwarding phase. The destination receives the message copies whenever it encounters with another node; or the destination retrieve the message copies by searching in visiting thrown box before this message is expired.

We present the U-thrown boxes, as shown in Algorithm 1. Algorithm 1 is a forwarding process after the spraying phase is already finished.

Algorithm 1 - Forwarding phase with strategy I

For each mobile node *i* do

If node *i* encounters another node *j* then

If node j is the destination then

Node *i* send the copy to *j*

If $U(j) > U(i)$ then

Node *i* forward the copies to node j

If node *i* visits thrown boxes then

Node *i* drops all its copies to thrown box

4.3 Thrown box Placement Schemes

To evaluate our proposed algorithm under different scenarios, we choose the appropriate thrown box locations for the deployment. Especially, when there is only a small amount of thrown boxes available to deploy in DTNs, these schemes play an important role in network performance. By analyzing social properties, we have m- most popular locations that we get from synthetic traces. For each mobile node, we denote its location by $L^j = \{l_1^j, l_2^j, ..., l_m^j\}$, which is a list of m locations visited by mobile node j. Notice that here each mobile node has the same list of locations which is potential to put thrown boxes. Thus a list of all locations in network is $L = \{l_1, l_2, ..., l_m\}$ .Then $\lambda(l_k^j)$ denotes the visiting probability of node j to location l_k. Thus we have the average visiting probability of all nodes to location l_k as follows:

$$\overline{\lambda}(l_k) = \frac{\sum_{j=1}^{n} \lambda(l_k^j)}{n} \quad (2)$$

Similarly, we also introduce the visiting duration, which denotes the average visiting duration that each mobile node reside in each location. Specifically, $t(l_k^j)$ is the average duration time that node j reside in location l_k during each time periods. We have the average visiting duration of all nodes to location l_k as follows:

$$\overline{t}(l_k) = \frac{\sum_{j=1}^{n} t(l_k^j)}{n} \quad (3)$$

Scheme 1: we simply choose a list of g locations called G with highest visiting probability and highest visiting duration. Among this location, mobile nodes visit most frequently and reside in these locations for long duration. We denote this list by G:

$$G \in L: \begin{Bmatrix} \overline{\lambda}(l_j) > \overline{\lambda}(l_k) \\ \overline{t}(l_j) > \overline{t}(l_k) \end{Bmatrix} \quad where: \begin{cases} l_j \in G \\ l_k \in (L \setminus G) \end{cases} \quad (4)$$

Scheme 2: This scheme has processed after a list G was chosen. Then we choose list of h

locations called H where nodes visit more frequently and reside in these locations for relative short duration.

$$H \in (L \setminus G) : \begin{Bmatrix} \overline{\lambda}(l_j) > \overline{\lambda}(l_k) \\ \overline{t}(l_j) < \overline{t}(l_k) \end{Bmatrix} \quad where : \begin{cases} l_j \in H \\ l_k \in (L \setminus H) \end{cases} \tag{5}$$

5. PERFORMANCE EVALUATION

In this section, we conduct extensive simulations to evaluate the performance of our proposed algorithm under various settings, as follows:

5.1 Simulation environments

In our simulation, we use the synthetic traces that are generated by a Time-Variant Community Model (TVCM) [18], to implement the thrown boxes into network area. This is because the general traces (e.g. Cambridge Haggle) do not provide adequately the community information. In contrast, TVCM has independent node mobility by design and we can modify the parameters suited with our network model as needed, which is useful to the performance evaluation of our algorithm.

In order to create the trace, in each simulation we deploy N=100 mobile nodes moving within a grid of 1500 x 1500, a square area composed of 17 square non-overlapping sub-areas, each of which represents a community area. We consider a sub-area in which nodes visit more frequently, is called high-visiting area, while the other areas are visit less frequently called a normal-visiting area. The visiting probability of each node, which is equal to $\lambda(b_i^j)$, is derived from Uniform distribution.

Settings	Value
Message generate interval	10-20 (seconds)
Message time-to-live	10 ~ 90 minutes
Message buffer-size	500kB
Number of copies	1 ~ 12 copies
Simulation time	12 hours, 27.8 hours
Number of nodes	100
Number of thrown boxes	1,5,10

The detailed simulation area is as follows:

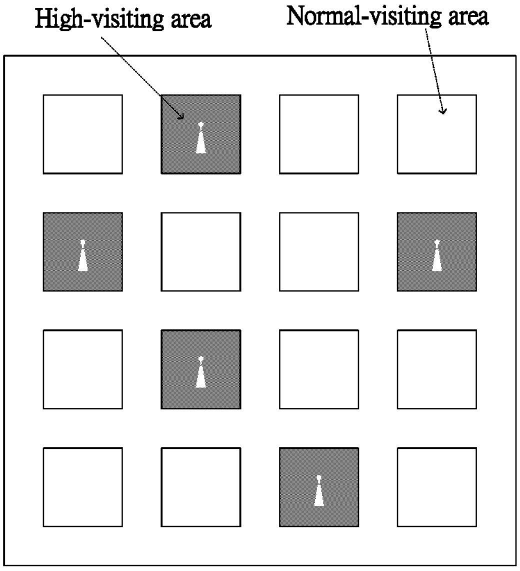

Figure 3: The simulation area

5.2 Simulation Settings and Metrics

To validate our simulation, we use The ONE [20] as the main simulator to evaluate the algorithms with the traces created from TVCM. Specifically, we compare our proposed algorithm (U-thrown box) with Binary Spray and Wait and Spray and Wait plus.

1) *Binary Spray and Wait*: this protocol acts in two phases. In the spray phase, as message with a replication quota is repeatedly copied to encountered nodes, and when the quota is decreased to one, a node enters to the wait phase and carries the message till it meets the destination

2) *Spray and Wait Plus*: acts as the same way as Spray and Wait. However, this protocol is assisted by extra stationary nodes which are deployed as the same locations as U-thrown box. The purpose of this protocol is to evaluate U-thrown box algorithm by comparing with another with the same amount of extra devices added in DTNs.

We adopt *Delivery probability* metrics to measure our proposed algorithm in our simulations. *Delivery probability* represents the ratio of the number of successful delivered messages over the number of created messages.

5.3 Evaluation Results

5.3.1 Performance in algorithms

We conduct two groups of simulation to evaluate the performance of U-Thrown box under two scenarios. Each group we adopt an appropriate thrown box placement scheme to maximize the network performance.

Group 1: in this scenario, mobile nodes visit frequently to g locations in which the total visiting probability ranged from 0.7 – 0.9, while other locations is visited rarely. In addition, the visiting duration of mobile nodes to these g locations is greater 3-4 times than other locations. Therefore, we can adapt scheme 1 to choose 5 appropriate locations for thrown box deployment. The results in fig. 4 show that the delivery probability of the three algorithms increase when the value of Time-to-live is increasing. Binary Spray and Wait has the worst delivery probability. Spray and wait plus is added with extra relay nodes to speed up message forwarding performance, has a medium performance. In contrast, the U-Thrown boxes, which is assisted by thrown boxes, exploits the utility function, has a better performance results among these algorithms.

Group 2: in this scenario, each mobile node tends to visit one location often with visiting probability ranging from 0.7 – 0.9 and stays in its location relatively long duration. There are h locations where the node visits quite often and stays for short duration. Therefore, we can adapt scheme 2 to choose 5 appropriate locations for thrown box deployment. The delivery probability results of three algorithms in fig. 5 is relatively low due to node's behavior. When mobile node tends to reside in its location, it has less opportunities to meet with another node and forward the copies to destination. However, U-Thrown box still achieve the highest performance among three algorithms.

5.3.2 Number of Thrown boxes

In order to study the effect of the number of deployed thrown boxes, we conduct another scenario. In this scenario, we have found that there are 10 high-visiting locations where mobile nodes frequently visits and stay for short duration. For the simulations, we vary the number of deployed thrown boxes from 1 to 10. The results in fig. 6 shows that the number of thrown boxes greatly influence the performance of delivery probability. Specifically, when the number of copies increases, the performance of U-Thrown box with 10 deployed thrown boxes increases significantly. While U-Thrown box with 1 deployed thrown box has low performance due to the limited number of thrown boxes. Moreover, a positive correlation was found between the forwarding performance and the deployed locations of thrown boxes. To increase the performance, the locations where we have found by the particular thrown boxes placement scheme, we need to put an adequate amount of thrown boxes.

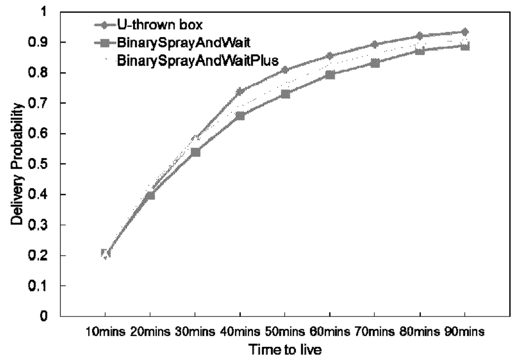

Figure 4: Performance comparison with n = 100, b = 5, simulation time = 12 hours

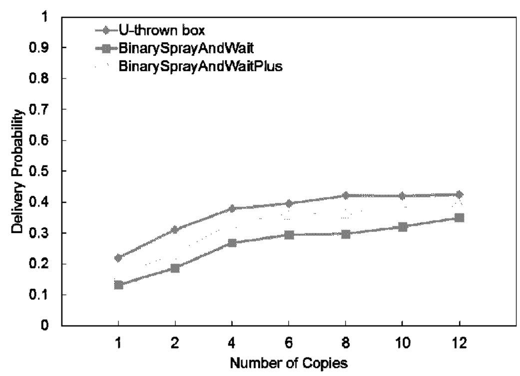

Figure 5: Performance comparison with n = 100, b = 5, simulation time = 27.8 hours

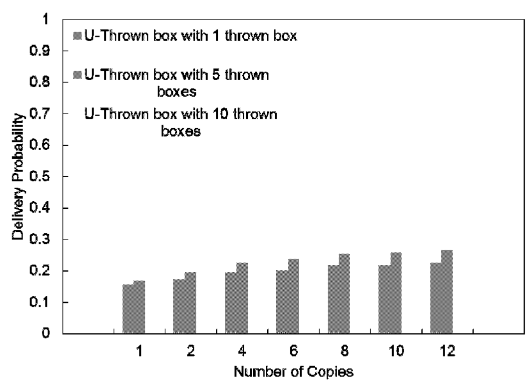

Figure 6: Performance comparison with N = 100, b = 5, simulation time = 27.8 hours

6. CONCLUSION

This paper studies the advantage of deploying "thrown boxes" as a stationary relay node for message forwarding efficiency in a network. The utility-based algorithm is also proposed to make the message forwarding decision when the contact between two nodes is established. In addition, simulation results show that the proposed algorithm achieves a considerable performance as the existing typical algorithms. Also, it is shown that thrown boxes play an important role in message forwarding process.

Acknowledgments

This work was supported in part by the National Science Council of Taiwan, R.O.C., under Contract NSC100-2221-E-008-085-MY3. We thank for Mr. Yu-Feng Hsu, Pin-Chun Chiu and Hau-Yun Lin for their supports in improving this manuscript.

REFERENCE

[1] C.E. Perkins, ed. AdHoc Networking, Addison-Wesley Inc., 2001.

[2] Zhensheng Zhang, "Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges," *Communications Surveys & Tutorials, IEEE* vol.8, no.1, First Quarter 2006.

[3] Zhao, Wenrui, et al. "Capacity enhancement using throwboxes in DTNs."*Mobile Adhoc and Sensor Systems (MASS), 2006 IEEE International Conference on.* IEEE, 2006.

[4] A. Balasubramanian, B. N. Levine, and A. Venkataramani. Replication routing in dtns: A resource allocation approach. IEEE/ACM Transactions on Networking, 18(2):596–609, 2010.

[5] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. Maxprop: Routing for vehicle-based disruption-tolerant networking. In Proceedings of IEEE INFOCOM'06, April 2006.

[6] X. Wang, Y. Shu, Z. Jin, Q. Pan, and B. S. Lee. Adaptive randomized epidemic routing for disruption tolerant networks. *In Proceedings of the 5th International Conference on Mobile Ad-hoc and Sensor Networks*, pages 424–429, December 2009

[7] Spyropoulos, Thrasyvoulos, Konstantinos Psounis, and Cauligi S. Raghavendra. "Spray and wait: an efficient routing scheme for intermittently connected mobile networks." In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking.* ACM, pages 252-259, 2005.

[8] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S. Raghavendra. 2007. Spray and Focus: Efficient Mobility-Assisted Routing for Heterogeneous and Correlated Mobility. In *Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications Workshops*, March 2007.

[9] A. Lindgren, A. Doria, and O. Schel´en. Probabilistic routing in intermittently connected networks. ACM SIGMOBILE Mobile Computing and Communications Review, 7(3):19–20, 2003.

[10] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "Maxprop: Routing for vehicle-based disruption tolerant networks," in Proceedings of IEEE INFOCOM, 2006, pp. 1–11.

[11] Elizabeth M. Daly and Mads Haahr. 2007. Social network analysis for routing in disconnected delay-tolerant MANETs. In *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing* (MobiHoc '07). ACM, New York, NY, USA, 32-40.

[12] Ibrahim, Mohuammad, Philippe Nain, and Iacopo Carreras. "Analysis of relay protocols for throwbox-equipped dtns." In *Proceedings of 7th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, 2009.*

[13] Muhammad Mukarram Bin Tariq, Mostafa Ammar, and Ellen Zegura. 2006. Message ferry route design for sparse ad hoc networks with mobile nodes. In *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing* (MobiHoc '06)

[14] Gu Bo, and Hong Xiaoyan. "Optimal routing strategy in thrownbox based delay tolerant network." In *Proceedings of 6th* IEEE *International ICST Conference on Communications and Networking in China (CHINACOM),* 2011.

[15] Gu Bo, and Hong Xiaoyan. "Capacity-aware routing using thrownboxes." In *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM),* 2011.

[16] Gu Bo, Hong Xiaoyan, and Wang Pu. "Analysis for bio-inspired thrown-box assisted message dissemination in delay tolerant networks." *Telecommunication Systems, vol. 52, no. 1, pp. 217-227, Jan 2013.*

[17] Hong Xiaoyan, Bo Gu, Yuguang Zeng, and Jingyuan Zhang. "Constructing time-varying contact graphs for heterogeneous delay tolerant networks." In *Proceedings of IEEE Global Communications Conference (GLOBECOM),* pp. 5302-5307, 2012.

[18] W. Hsu, T. Spyropoulos, K. Psounis, and A. Helmy. Modeling timevariant user mobility in wireless mobile networks. In Proceedings of INFOCOM 2007, Anchorage, AL, May 2007.

[19] Xiaolan Zhang, Jim Kurose, Brian Neil Levine, Don Towsley, and Honggang Zhang. 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

[20] Ari Keränen, Jörg Ott, and Teemu Kärkkäinen.The ONE simulator for DTN protocol evaluation. In *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, 2009