An Improved Spray and Wait Algorithm based on RVNS in Delay Tolerant Mobile Sensor Networks

Kun Wang[1, 3], Yun Shao[1], Lei Shu[2], Yanfei Sun[1], and Lei He[3]

[1]Nanjing University of Posts and Telecommunications, Nanjing, 210003, China. Emails: kwang@njupt.edu.cn, shaoyunn2848@qq.com, njsyf@vip.163.com
[2]Guangdong University of Petrochemical Technology, Guangdong, 525000, China. Email: lei.shu@live.ie
[3]University of California, Los Angeles, 90095, USA. Emails: lhe@ee.ucla.edu

***Abstract*—Due to the limited resources of DTMSN (Delay Tolerant Mobile Sensor Networks), network congestion becomes a critical problem to resolve. Traditional congestion control methods where the number of copies is restricted to limit data packet forwarding cannot adapt to constantly changing network environment because of fixed number of copies. Fortunately, this problem can be solved through a real-time algorithm by modifying data packet forwarding conditions. However, one of the major challenges of this algorithm is detecting characteristics of the network environment accurately and efficiently. In this paper, an optimized routing algorithm, RVNS (Reduced Variable Neighborhood Search)-based Spray and Wait (SW) is proposed. In this algorithm, nodes will transmit and store the counter record of each other when they meet, based on which, RVNS is introduced to calculate a real-time threshold for the forwarding condition to control packet delivery. Simulation results show that the proposed algorithm increases delivery probability and dramatically reduces the overhead ratio. In some extreme cases, this algorithm can reach an extremely low overhead ratio (ten times lower than that of SW), meaning that the proposed algorithm suits challenged networks well.**

Keywords*—*Reduced Variable Neighborhood Search (RVNS), Spray and Wait, Routing Algorithm, Delay Tolerant Mobile Sensor Networks (DTMSN), Congestion Control, Redundant Copies

I. INTRODUCTION

AMONG many basic routing algorithms of DTMSN (Delay Tolerant Mobile Sensor Network) like Direct Transmission algorithm (DT) [1], Flooding algorithm [2], Epidemic algorithm [3], PROPHET algorithm [4], one of the most important issues is their high overhead and redundant copies of packets, especially in Flooding and Epidemic. To overcome this problem, T. Spyropoulos et al. designed the limited-flooding-based SW (Spray and Wait) [5], in which packets are assigned with limited copies. SW algorithm includes two stages: Spray stage and Wait stage. In Spray stage, a source node will generate a packet with L copies, then these copies will be delivered to L different relaying nodes. In Wait stage, these L relaying nodes will execute DT if none of these L nodes is the destination.

However, Since SW is based on limited flooding, it remains a problem that large number of copies generated in Spray stage will occupy the limited buffer space, as shown in Fig. 1. In Fig. 1 (a), the surrounding nodes' buffer tends to be filled up. However, according to the transmission scheme in Spray stage, V_i will still deliver the its stored packets (represented in purple) to its neighbors so that they have to drop several packets, as shown in Fig. 1 (b). To store the received packets (represented in green), lots of packets in surrounding nodes are dropped (represented in yellow), which makes network environment become awful, and is hard to store newly generated packets (represented in red).

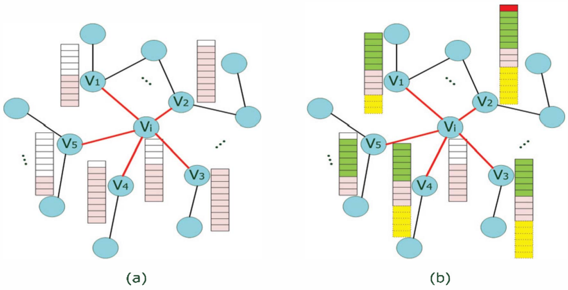


Fig. 1: (a) Buffer status before transmission; (b) Buffer status after transmission.

Under this condition, it is a key issue to be concerned that how nodes buffer status can be estimated timely in a specific area. To this end, we adopt RVNS to modify the forwarding mechanism in Spray stage, and optimize the buffer utilization of network [6]. The major contributions of our work are summarized as follows.

- To the best of our knowledge, it is the first time to apply RVNS into congestion control of packet delivery in routing algorithms. Generally, RVNS is usually used to solve combinational optimization problems by outputting a maximum or minimum value. But proposed algorithm returns a intermediate value through controlling the number of neighborhoods reconstructions;
- This paper designs a new mechanism for congestion control in Spray stage in SW. This stage is optimized through a real-time congestion threshold to reflect network congestion condition. Then, the forwarding decision in Spray stage is made based on this congestion

threshold and available buffer so that congestion decreases.

The rest of this paper is organized as follows. The existing related works of SW are introduced in Section II. Section III presents the basic idea of RVNS. The description and analysis of main algorithm are shown in Section IV. Section V demonstrates the performance evaluation and result analysis and Section VI concludes the paper.

II. RELATED WORK

At present, the optimizing mechanism of SW can be divided into following categories:

(1) Optimizing Spray stage

A. Al-Hinai et al. proposed a new routing algorithm TB-SnW (Trust-based Spray-and-Wait). In TB-SnW, each node maintains a trust level of meeting nodes based on forwarding history, and distributes packets based on trust level to avoid black hole attack [7]. E. Bulut et al. proposed a multiperiod spraying algorithm. In this algorithm, copies will be forwarded into network periodically, and the total number of copies will be decided by the urgency of message. Due to only small number of copies are forwarded in the initial period, this algorithm can minimize the average copies [8].

To improve delivery probability, the algorithms above usually adopt Epidemic algorithm to spread copies in a flooding way, which will cause the competition of network resource and increase energy consumption.

(2) Optimizing Wait stage

T. Spyropoulos et al. proposed an improved SF (Spray and Focus) algorithm. Different from SW, an utility function representing historical meeting time is introduced in Wait stage of SF to help nodes deliver the only copy to others with higher utility [9]. S. M. Iqbal et al. proposed a different condition entering Wait stage. This entering condition will be activated when TTL (Time To Live) decreases to a set value [10].

These improvements of Wait stage mainly focus on the optimization of DT. Delivery delay is largely shortened. However, congestion issue is still not considered.

(3) Optimizing other aspects

N. Kishore et al. proposed a multi-copies routing algorithm in which replication is controlled to decrease the overhead ratio through the scalability evaluation and buffer space of nodes in different network scale [11]. C. G. Requena et al. proposed a distributed routing methodology for fat trees. It adopts a mechanism to detect exclusion intervals with fault, and allows forwarding happens through healthy paths to avoid the influence of network failures [12].

In terms of the researches above, nodes' buffer status and network congestion have not been considered. Although the number of copies has been fixed when packets are generated, it cannot avoid the great amount of copies forwarded in network and the occupation of massive buffer space. Accordingly, a RVNS-based algorithm optimizing Spray stage is proposed to solve the problem of congestion.

III. INTRODUCTION OF RVNS ALGORITHM

In this section, RVNS algorithm and its original algorithm Variable Neighborhood Search will be introduced.

RVNS is a simplified version of VNS. In RVNS, there is a solution space S (S=x_1,x_2,x_3,...,x_n), where initial solution will be obtained. For optimization problems, if $\forall x* \in S$ satisfies $f(x*) \leqslant f(x)$, solution $x*$ will be regarded as a new feasible global optimal solution and replace old global optimal solution x, where $f(x)$ denotes a utility function. Correspondingly, for a global optimal solution x, there is a neighborhood structure $N(x)$ based on x. Just as $x \in S$, $N(x)$ is a subset of S, namely, $N(x) \subseteq S$. In a neighborhood structure with k partitioned neighborhoods, these specific neighborhoods are denoted as N_k, where $k \in (1,k_{max})$. Particularly, when x is a feasible global optimal solution, these k specific neighborhoods are denoted as $N_k(x)$. N_k will generate different neighborhood structures through a series of matrix transformations for next iteration if a new feasible global optimal solution $x*$ has been found.

Different from global optimal solution $x*$ (global minimum reached by optimization), local optimal solution is denoted as x' ($x' \in S$, a local minimum reached by optimization), namely, we cannot find another x satisfying $f(x) < f(x')$. In RVNS, x' will be generated randomly in $N_k(x)$. When x' satisfies $f(x') < f(x)$, the value of k will be returned, and the algorithm will reconstruct neighborhood according to x'. Otherwise, it will switch to next neighborhood.

Currently, RVNS-based meta-heuristic algorithms are always adopted to LRP (Location Routing Problem) [6]. Therefore, it is a new attempt to introduce this algorithm to the congestion control of DTMSN routing.

IV. RVNS-BASED SPRAY AND WAIT (RSW)

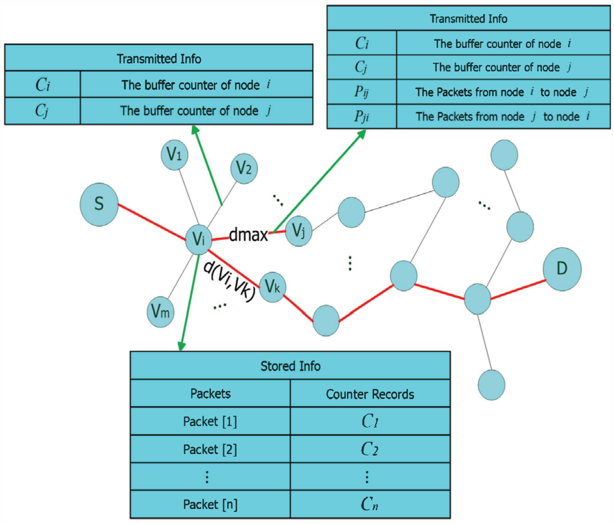
The proposed algorithm RSW introduces RVNS to adjust the forwarding mechanism of packets according to nearby network environment of nodes. In the improved algorithm, a congestion threshold is calculated through RVNS. This threshold will decide whether the packet is forwarded as original SW or wait for another forwarding opportunity.

A. Network Model

All nodes (sensors) in DTMSN are arbitrarily deployed in an abstract network model as a graph $G = (V,E)$, where V (V=v_i,v_2,v_3,...,v_n) represents n nodes moving in a two-dimensional rectangular area. Every node will move towards different direction with different speed, varying between [V_{min},V_{max}]. In this paper, we assume all the nodes share the same radio communication range. When the distance between two nodes is shorter than this radio communication range, a communication link will be set up. $E = \{v_i,v_j | d(v_i,v_j) < d_{max},v_i,v_j \in V\}$ represents the total communication links in the network, where d_{max} represents the maximum radio communication range. In every node, an additional buffer is defined to store buffer statues records of other nodes. The network model is shown in Fig. 2.

Besides the model description above, other assumptions are listed as follows:

(1) Every node owns a buffer counter C_i ($i \in [1,n]$), which represents the number of stored packets in node i, and we use C_{imax} to indicate that the buffer space of node i is full.



Transmitted Info	
C_i	The buffer counter of node i
C_j	The buffer counter of node j

Transmitted Info	
C_i	The buffer counter of node i
C_j	The buffer counter of node j
P_{ij}	The Packets from node i to node j
P_{ji}	The Packets from node j to node i

Stored Info	
Packets	Counter Records
Packet [1]	C_1
Packet [2]	C_2
⋮	⋮
Packet [n]	C_n

Fig. 2: Network model

(2) The buffer counter of nodes will be recorded successively in buffer. For example, if the latest three meeting nodes of node i are in sequence of node a, b and c, the buffer space of node i will store C_a, C_b and C_c in sequence.

(3) When two nodes meet, they will exchange and store each other's buffer counter.

B. Buffer Status Level

As described above, every node has a specific space to store buffer counter of meeting nodes.

Definition 1: $M_i(i \in [1,n])$ is a space in node i recording buffer counter, and m is the maximum capacity of this space. $M_i(h)(i \in [1,n], h \in [1,m])$ represents the h^{th} record in node i. According to this definition, when node a meets node b, one counter record will be added to both M_a and M_b, namely, C_b and C_a.

Definition 2: A new concept *buffer status level* is defined here to indicate the network status around a specific node. For example, if we divide the buffer status into three levels, $l(1)$, $l(2)$ and $l(3)$ will stand for idle, medium and congestion, respectively. So we use $l(max)$ to denote the highest level. By C_{imax}, the minimum and maximum values can be calculated for every level $l(i)(i \in [1,max])$ as follows:

$$l(i)_{min} = (i-1) \times \frac{C_{imax}}{max} + 1 \tag{1}$$

$$l(i)_{max} = i \times \frac{C_{imax}}{max} \tag{2}$$

where max is the number of buffer status level for stored counter records. Since every stored record will be divided into a specific level, massive divisions can reflect the congestion level of whole network. For example, if lots of records are classified into high levels, it means the network congests. Oppositely, if many records are classified into low levels, network is relatively idle.

C. Algorithm Description

In this section, RVNS is used to find out a threshold, denoted as congestion threshold, to indicate the congestion level of the whole network. According to traditional SW, if the buffer space of a node is full, lots of packets are waiting to be delivered in Spray stage when it meets other nodes. At the same time, large buffer spaces are also required at the receiving node to store packets. Once the buffer of receiving node is also full, many packets to be forwarded have to be discarded, which consumes the nodes' energy. Accordingly, a controlling mechanism based on RVNS is designed to save energy.

1) ***RVNS implementation:*** In the algorithm, the counter records in buffer space are served as solution space. An initial solution is assigned randomly. In RVNS, we construct neighborhood structure according to the level of every record and current global optimal solution. After several times of search and neighborhood reconstruction, an accurate parameter can be obtained to reflect current network environment.

Definition 3: $S_i = \{M_i(1), M_i(2), M_i(3), ..., M_i(m)\}(i \in [1,n])$ is a solution space for RVNS in node i, namely, a set of counter records in buffer space.

Definition 4: $M_i(x)(i \in [1,n], x \in [1,m])$ is a counter record (current global optimal solution) of RVNS in node i. $N_k(x)(k \in [1,k_{max}])$ is the neighborhood structure constructed by $M_i(x)$, where k_{max} is the number of neighborhood structure. Specifically, assuming that the h^{th} counter records in node i $M_i(h)(i \in [1,n], h \in [1,m])$ belongs to level $l(temp)(temp \in [1,max])$ and global optimal solution $M_i(x)(i \in [1,n], x \in [1,m])$ belongs to level $l(opt)(opt \in [1,max])$, then record $M_i(h)$ can be divided into the k^{th} neighborhood constructed by $M_i(x)$, as shown in Fig. 3, where

$$k = |l(temp) - l(opt)| \tag{3}$$

In Fig. 3, x in yellow box represents the index of current global optimal solution and h (increasing from 1 to m) in green box represents the index of record being analyzed now. The buffer status level of $M_i(x)$ and $M_i(h)$ belong to $l(opt)$ and $l(temp)$, respectively. Afterwards, which neighborhood should $M_i(h)$ be divided into can be calculated, as shown in green neighborhood.

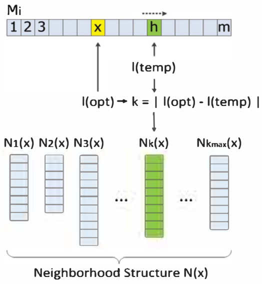


Fig. 3: Neighborhood division in Definition 4

Before every iteration of RVNS, each record is divided into corresponding neighborhood according to current global optimal solution x. When RVNS begins, a random point x'

from the first neighborhood $(N_1(x))$ is selected. If $x' < x$, x' is set as current global optimal solution and then neighborhood structure is reconstructed. Otherwise, the algorithm switches to next neighborhood to continue search process. This process is replicated until the number of neighborhood reconstruction achieves to maximum value t_{max} (iterations of RVNS).

2) ***Packets delivery Mechanism:*** After several times of search and neighborhood reconstruction, RVNS can return a suitable value $M_i(opt)(opt \in [1,m])$, which can reflect current network environment. Larger $M_i(opt)$ indicates more congested level of network. Smaller $M_i(opt)$ means that network is idler. At the same time, it is notable that restriction needs to be carefully selected. Too stringent restriction will totally block packets delivery, so that new-generated packets cannot be forwarded. To prove this thought, we take $M_i(opt)$ as congestion threshold T_i to constrain forwarding process, namely, $T_i = M_i(opt)$. When node i meets node j, node i compares T_i with C_j. If $C_j > T_i$, it means buffer space in node j is too full to receive new packets. In order to avoid congestion and packets loss, even though some packets in node i are in Spray stage, no forwarding will happen from node i to node j. Specifically, when T_i is too small, it may block most of packet delivery. Accordingly, a lower boundry b_l is set to avoid this circumstance. In this paper, b_l is set to be $C_{imax}/2$.

V. PERFORMANCE EVALUATION

In this section, the performance of RSW is verified through two parts' simulations. Simulation configuration and performance evaluation are implemented in C++. In this simulation environment, random movement of nodes in network can be realized, and performance results by proposed algorithms can be obtained. The whole simulation is set in a 100m*100m two-dimensional region. There are [100, 1000] mobile nodes deployed in this area. They move towards different directions with speeds varying from 0 to 4m/s. Radio communication range is set as [3m, 10m]. Those records are divided into 4 levels. Packet size is 50KB. During all the simulations, there are 100 packets generated, and simulation time is set 2000s.

The following three performance metrics are used to evaluate the performance of our algorithm.

1) $Delivery\ probability = delivered\ messages/total\ messages = delivered/created.$

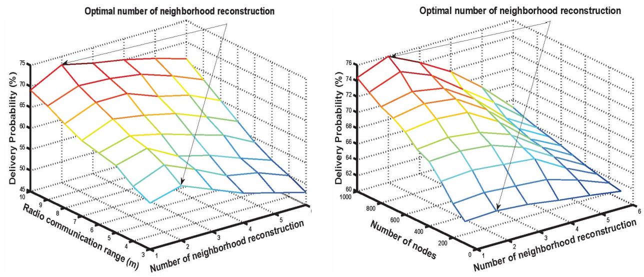
2) $Overhead\ ratio = forwarded\ but\ undelivered\ messages/delivered\ messages = (relayed - delivered)/delivered.$

3) $Average\ latency = \sum(receiving\ time\ of\ message\ i - sending\ time\ of\ message\ i)/delivered\ messages = \sum(tir - tis)/delivered.$

A. Algorithm parameters selection

As mentioned before, ensure accuracy of congestion threshold, number of neighborhood reconstruction should be restricted. In this section, we design simulation to find out the optimal number of neighborhood reconstruction with different radio communication range and number of nodes.

In the simulation, we observe the changes of delivery probability with the number of neighborhood reconstruction under different simulation environment. The simulation results are shown in Fig. 4. In Fig. 4 (a), the number of node is 100, and in Fig. 4 (b), the radio communication range is 7m. As shown in Fig. 4, delivery probability changes with the different number of neighborhood reconstruction under different circumstances. It is because improper number will lead to inaccurate value of congestion threshold. Simulation results demonstrate that setting the number of neighborhood reconstruction as two can achieve the best algorithm performance.



(a)

(b)

Fig. 4: Number of neighborhood reconstruction

B. Algorithm Comparison

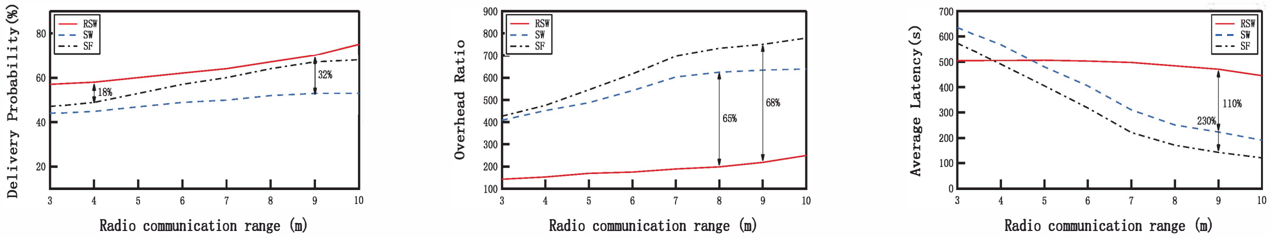
In this section, we compare RSW with SW and SF to observe the change of three performance metrics with different value of radio communication range and number of nodes.

(1) There are 100 nodes deployed in the network. Buffer space is set as 500K. Simulation time is 2000s and there are 100 packets generated during this period of time. The radio communication range is set as 3m, 4m, 5m, 6m, 7m, 8m, 9m and 10m, respectively. Simulation results are shown in Fig. 5.

As shown in Fig. 5, the improvement of RSW is smaller than that of SW and SF with larger latency and smaller increase of delivery probability, but the overhead ratio of RSW is still superior. It is because the congestion threshold may constrain the meaningless forwarding to some extent. The delivery probability of RSW is 41.51% and 21.28% higher than SW and SF at most, and the overhead ratio is only 33.66% and 32.06% of SW and SF at least.

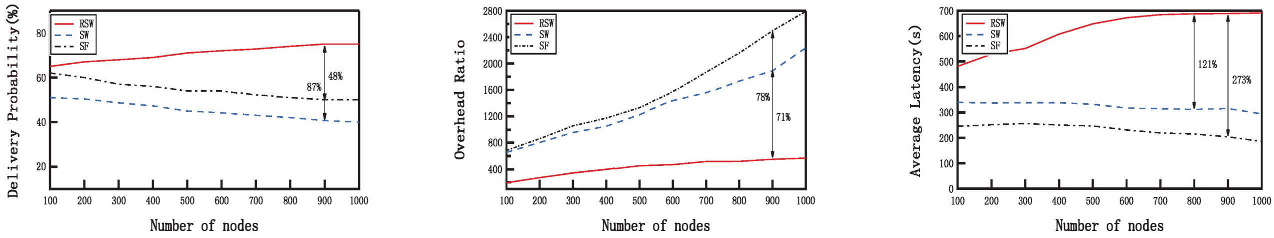
(2) Radio communication range is set to 7m and buffer space to 500K. Simulation time is 2000s and there are 100 packets generated during this period of time. The number of nodes is set to 100, 200, 300, 400, 500, 600, 700, 800, 900 and 1000, respectively. Simulation results are shown in Fig. 6.

With the number of nodes increasing, the meeting between nodes becomes more frequently. Under this circumstance, many packets are forwarded in an area frequently even though the buffer spaces in this area are filled up. However, due to congestion threshold In RSW, forwarding will be paused when a node detects that its surrounding nodes buffer space are filled up, and recovered when environment is suitable, which optimize delivery probability and overhead ratio. As shown in Fig. 6, the delivery probability of RSW is 87.46% and 48.83% higher than SW and SF at most, and the overhead ratio is only 24.96% and 20.38% of SW and SF at least.



(a) (b) (c)

Fig. 5: (a) Delivery probability, (b) Overhead ratio, and (c) Average latency changes with radio communication range, respectively.



(a) (b) (c)

Fig. 6: (a) Delivery probability, (b) Overhead ratio and (c) Average latency changes with number of nodes, respectively.

VI. CONCLUSION

In this paper, an optimized RVNS-based Spray and Wait is proposed where RVNS uses the counter record in buffer space to indirectly reflect current network environment and meaningless forwarding is controlled according to a real-time congestion threshold. Besides, the variation in different time and different areas in DTMSN has been fully considered in proposed algorithm. Although further improvement are needed in the aspects of RVNS invoking and congestion threshold calculation, RSW is verified to be an efficient strategy in challenged networks.

ACKNOWLEDGMENT

The authors would like to thank NSFC (61100213, 61170276, 61201160, 61201222, 61401107); Specialized Fund for the Doctoral Program of Higher Education (20113223120007); Natural Science Foundation of Jiangsu Province (BK20141427); Huawei Innovation Research Program (YB2014010048).

REFERENCES

[1] M. Sharma and A K. Shaw, Transmission Time and Throughput Analysis of EEE LEACH, LEACH and Direct Transmission Protocol: A Simulation Based Approach. Advanced Computing, Vol. 3, No. 6, pp. 75-82, 2012.

[2] K. Natesapillai, V. Palanisamy, and K. Duraiswamy, Optimum Density based Model for Probabilistic Flooding Protocol in Mobile ad hoc Network. European Journal of Scientific Research, Vol. 39, No. 4, pp. 577-588, 2010.

[3] X. Zhang, G. Neglia, and J. Kurose, Performance Modeling of Epidemic Routing. Computer Networks, Vol. 51, No. 10, pp.2867-2891, 2007.

[4] L. Anders, D. Avri, and S. Olov, Probabilistic Routing in Intermittently Connected Networks. ACM SIGMOBILE Mobile Computing and Communications Review, Vol. 7, No. 3, pp. 19-20, 2003.

[5] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks. in Proc. ACM SIGCOMM workshop, 2005, pp. 252-259.

[6] P. Hansen, N. Mladenovic, and J. A. Moreno, Variable Neighbourhood Search: Methods and Applications. Journal of Operations Research, Vol. 6, No. 4, pp. 319-360, 2008.

[7] A. Al-Hinai, H. Zhang, and Y. Chen, TB-SnW: Trust-based Spray-and-Wait routing for Delay-Tolerant Networks. Journal of Supercomputing, Vol. 10, No. 2, pp.1-17, 2014.

[8] E. Bulut, Z. Wang, and B. Szymanski, Cost Effective Multi-Period Spraying for Routing in Delay Tolerant Networks. IEEE Trans. Networking, vol. 18, no. 5, pp. 1530-1543, 2010.

[9] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, Spray and Focus: Efficient Mobility-assisted Routing for Heterogeneous and Correlated Mobility. in Proc. of 5th Annual IEEE International Conference on PerCom Workshops, 2007, pp. 79-85.

[10] S. M. A. Iqbal and A. K. Chowdhury, Adaptation of Spray Phase to Improve the Binary Spray and Wait Routing in Delay Tolerant Networks. in Proc. of ICCIT, 2012, pp. 261-266.

[11] N. Kishore, S. Jain, and V. Soares, An Empirical Review on the Spray and Wait Based Algorithms for Controlled Replication Forwarding in Delay Tolerant Networks. in Proc. of WOCN, 2013, pp. 1-5.

[12] C. G. Requena, M. E. G. Requena, P. J. L. Rodriguez, and J. F. D. Marin, FT2EI: A Dynamic Fault-Tolerant Routing Methodology for Fat Trees with Exclusion Intervals. IEEE Trans. Parallel and Distributed Systems, vol. 20, no. 6, pp. 802-817, 2009.