2012

A Joint Replication-Migration-based Routing in Delay Tolerant Networks

Yunsheng Wang
Temple University

Jie Wu
Temple University

Zhen Jiang
West Chester University of Pennsylvania, zjiang@wcupa.edu

Feng Li
IUPUI, Indianapolis, IN

Recommended Citation

Wang, Y., Wu, J., Jiang, Z., & Li, F. (2012). A Joint Replication-Migration-based Routing in Delay Tolerant Networks. *2012 IEEE Conference on Communications*, 73-77. http://dx.doi.org/10.1109/ICC.2012.6363990

A Joint Replication-Migration-based Routing in Delay Tolerant Networks

Yunsheng Wang and Jie Wu
Dept. of Computer and Info. Sciences
Temple University
Philadelphia, PA 19122

Zhen Jiang
Dept. of Computer Science
West Chester University
West Chester, PA 19382

Feng Li
School of Eng. and Tech.
IUPUI
Indianapolis, IN 46202

***Abstract*—Delay tolerant networks (DTNs) use mobility-assisted routing, where nodes carry, store, and forward data to each other in order to overcome the intermittent connectivity and limited network capacity of this type of network. In this paper, we propose a routing protocol that includes two mechanisms: *message replication* and *message migration*. Each mechanism has two steps: *message selection* and *node selection*. In message replication, we choose the smallest hop-count message to replicate. The hop-count threshold is used to control the replication speed. We propose a metric called *2-hop activity level* to measure the relay node's transmission capacity, which is used in node selection. Our protocol includes a novel message migration policy that is used to overcome the limited buffer space and bandwidth of DTN nodes. We validate our protocol via extensive simulation experiments; we use a combination of synthetic and real mobility traces.**

***Index Terms*—Buffer management, delay tolerant networks (DTNs), message migration, message replication, routing.**

I. INTRODUCTION

Delay tolerant networks (DTNs) allow for data to be transferred when mobile nodes encounter each other intermittently. There is no end-to-end path between some or all of the nodes in DTNs. There exist several different application scenarios: connectivity of developing countries [1], mobile social networks [2], and vehicular DTN road communications [3]. Intermittent connectivity, limited network capacity, storage and energy constraints, and the uncertainty of mobility patterns make routing in DTNs a challenging problem.

Several DTN routing schemes have been proposed recently [2–8]. The researchers investigated various routing protocols: flooding-based approaches [4, 5], encounter history-based approaches [3, 6, 7], and social behavior-based approaches [2, 8]. In this paper, we propose a joint replication-migration-based routing scheme that includes *message replication*: replicating the message to a relay node and *message migration*: migrating the message to an alternative node with enough buffer space when the current buffer is almost full.

In this paper, we use a metric, *2-hop activity level*, to control message replication during a contact. The notion of the activity level is based on the observation that an active node has a higher chance of contacting more nodes later to improve its performance. We use 2-hop neighborhood information (or simply 2-hop information) to predict the node's activity level, which combines local information and the encountered nodes' information. Our message replication scheme is an encounter history-based multiple copy model that considers the 2-hop information so that destinations can be reached quickly with a high delivery rate. There are two steps controlling the message replication: *message selection* (selecting the highest priority message to replicate) and *node selection* (selecting the best relay node to carry, store, and forward the message).

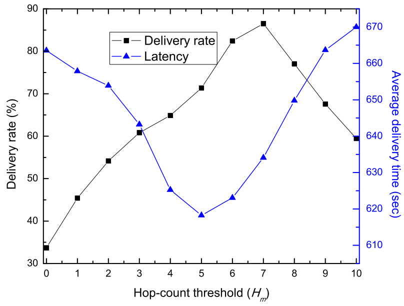


Fig. 1. Comparison delivery rate and latency in different hop-count thresholds using message replication.

We use message hop-count to prioritize the messages in the buffer. The *hop-count threshold* (T_h) is used to control the message replication speed. In other words, if the hop-count of the selected message is smaller than the threshold, the message will be replicated directly. Otherwise, we will use the node activity level to select a good relay node to replicate. However, when the threshold is low, the replication speed is slow. Hence, it has a higher latency. When the threshold is high, it performs significant replications, which may lead to buffer congestion when the buffer space is limited. As can be seen in Fig. 1, the delivery rate and latency functions are both quadratic. We find that the optimal hop-count thresholds (T_h) are different for maximizing the delivery rate or minimizing the latency in a synthetic trace of 20 nodes. Each of these nodes has a 10-message buffer.

Although a large amount of effort has been invested in the design of buffer management policies for DTNs [9–11], most of them deal with choosing the appropriate message to discard when the available buffer space is below a threshold (T_b). In this paper, we propose a novel congestion control scheme, called *message migration*, that supports early migration to an alternative node when the available buffer space is below T_b. The message migration policy has two parts: *message selection* (selecting the lowest priority message to migrate) and *node selection* (selecting an alternative node to store the message by considering the available buffer space in the alternative node). By using the message migration scheme, the message can be migrated from an overloaded node to an alternative node early which can increase the delivery rate with a cost of increasing the number of forwardings.

The major contributions of our work are as follows: (1) We present a joint replication-migration-based routing scheme to guide the messages to the destinations and to minimize latency with a high delivery rate. (2) We propose the notion of activity level, which is based on 2-hop neighborhood information. (3) We study the impact of the hop count threshold to control the message replication speed. (4) We propose a message migration process that migrates the message from the overloaded node to an alternative node to reduce message loss. (5) We evaluate the proposed scheme not only in synthetic traces, but also in real traces. The simulation results show the good performance of our protocol in DTN routing.

II. RELATED WORK

Routing in DTNs has attracted the attention of the research community recently. The most simple replication-based approach is flooding, also known as epidemic routing [4]. To control the copies of the message in epidemic routing, Spyropoulos et al. proposed the *spray and wait* algorithm [5] and the *spray and forcus* algorithm [12]. In the former one, the source node sprays the message to the neighbors and waits for one of the neighbors to meet the destination. The latter one goes further to allow multi-hop even when there is one copy, which is based on a quality metric from the nodes' encounter history. In this paper, our approach is also a form of replication-based routing, which considers 2-hop information to improve the accuracy. Our scheme combines message selection and node selection to improve the performance.

Several solutions have been proposed to handle storage congestion control problems in DTNs. Most of them are based on message dropping policies [9–11]. In [11], Seligman et al. investigated *storage routing* to avoid storage congestion in DTNs. The basic idea is that once a node becomes congested, it forwards some fraction of its stored messages to alternative custodians in order to decongest itself; therefore, the node will be able to serve incoming traffic. Our proposed message migration scheme has a similar spirit to that of [11], but we do not rely on global information. Our message migration policy is made using local information only.

III. REPLICATION-MIGRATION-BASED ROUTING

A. Objective and System Model

The objective of this paper is to develop an efficient routing scheme in DTNs by considering buffer congestion. Our approach is a joint replication-migration-based routing scheme that is based on 2-hop information. We consider the limited buffer space scenario in DTNs. Our protocol is based on prioritizing the schedule of the messages being transmitted to other nodes when there are multiple messages in the buffer based on message hop-count. Four performance metrics are used: (1) *delivery rate*: the rate of messages that reach the proposed destination within a given time limit; (2) *number of forwardings*: the average number of forwardings for each message in the entire routing process. This can be considered as the overhead for the routing process; (3) *latency*: the average duration between the generation time and arrival time of a packet; (4) *number of lost messages*: the number of messages that have been dropped.

We assume that there are N nodes in the whole network. M messages will be generated. Each of the messages has a message ID. Each of these nodes has a buffer in which it can store up to F messages in-transit: either messages belonging to other nodes or messages generated by themselves. Each node has a priority ranking table for the messages stored in its buffer based on the message hop-count.

In this paper, we consider the 802.11b transmitter that has a maximum data rate of 11Mbit/s. We assume that the average transfer opportunity duration is about 10 seconds. Hence, during each contact, about 10MB of data can be exchanged. We suppose that the message size is 1MB. Therefore, about 10 messages can be exchanged. To simplify our discussion, in one contact, we use one message to represent 10 messages to explain our approaches in the rest of the paper.

In the following, we present the details of the protocol: message selection and message migration. When nodes a and b come in contact with each other, if b is the destination of the message in the buffer of a, a will forward the message to node b. In the remaining part of this section, we only consider the scenario where b is not a destination.

B. Message Replication

1) Replication message selection: in our message selection stage, we use *hop-count* (H_m) [1] as the priority metric to select the smallest hop-count message m as the candidate replication message in a contact. If there are two messages with the same hop-count, our selection is based on the message ID.

To reduce the message delivery time, we propose a hop-count threshold (T_h) to control the message replication speed. If the hop-count of the selected message is smaller than T_h, we believe that it has not traveled far in the network. Hence, we will replicate this message to the relay node without node selection. In this way, the newer messages are replicated at several transfer opportunities when they are new, increasing their opportunities to reach the destinations. Otherwise, if the hop-count is larger than T_h, we will go to the replication node selection stage to decide whether to replicate the message to the encountered node.

2) Replication node selection: the node selection is based on the 2-hop activity level. Next, we present the definition of the 2-hop activity level. The 2-hop activity level of a node is the combination of its own encounter history and its neighbors' encounter histories before this contact. Node a's activity level (A_a) can be calculated as follows:

$$A_a = cE_a + (1-c)\sum_{k=1}^{i} w_k E_k, \tag{1}$$

where E_a is node a's total number of contact times [2] with other nodes in the network. w_k is neighbor k's contribution to a's activity level, which is the ratio of k that appeared in

---

[1] Hop-count is the number of forwardings that the message m has made.

[2] Although contact duration is also important, results in [13] show that there are high correlation coefficients of contact duration and contact times in many traces; we simply consider only contact times here.

node a's encounter list. c is a constant value to present the weight of the neighbors' contributions. For example, node a has encountered two other nodes, x and y, 2 and 3 times, respectively. By the definition, $E_a = 5$, and we assume that $E_x = 8$ and $E_y = 6$. If we set c to 0.8, and by using Eq. (1), we can get the activity level of a: $A_a = 0.8 \times 5 + 0.2 \times (\frac{2}{5} \times 8 + \frac{3}{5} \times 6) = 4.68$.

When nodes a and b are in each other's communication range, if the hop-count of the selected message m (H_m) is smaller than T_h, a replicates m to b. Otherwise, they will compare their activity levels. If node b's activity level A_b is larger than node a's activity level A_a, a replicates m to b. Otherwise, a replicates m to b with probability A_b/A_a. If b's buffer is full, we discard the largest hop-count message in the buffer of b, including the replicated message from a. The message replication process is shown in Algorithm 1 when nodes a and b are in contact.

Algorithm 1 Message Replication (at node a)

```
Nodes $a$ and $b$ first exchange the metadata.
$a$ selects the smallest hop-count message ($m$) to replicate.
if $H_m < T_h$ then
    $a$ replicates $m$ to $b$.
else if $A_b > A_a$ then
    $a$ replicates $m$ to $b$.
else $a$ replicates $m$ to $b$ with probability $A_b/A_a$.
```

C. Message Migration

Although there are many works that have designed message dropping policies, we believe that using message dropping policies will reduce the delivery rate in the following situation: if the buffer size is relatively small, the buffer of an active node will be overloaded quickly. Then, the messages in the buffer may all have a relatively small hop-count. Hence, dropping any message in the buffer will decrease the delivery rate.

Therefore, we propose a message migration scheme to control the buffer congestion. When the available buffer space is below a threshold ($S_a < T_b$), the node will migrate a message to an alternative node that has enough buffer space early on, even though it may not be a high priority node. In the message migration process, the sending node will forward the largest hop-count message to the alternative node and will not retain a copy in its buffer.

1) Migration message selection: we select the largest hop-count message to migrate. This is because a larger hop-count message has a smaller opportunity to be replicated in the congestion node.

2) Migration node selection: in order to make sure that the message will not be discarded quickly after the migration, the selected migration message will be migrated to an alternative node with enough buffer space ($S_b - 1 \geq T_b$).

When the available buffer space of node a (S_a) is below the buffer space threshold T_b in the message dropping schemes, m will be discarded to release the buffer space for new messages. However, it will reduce the delivery probability to the destination and will increase the delivery time because of the message loss. By using the message migration scheme, node a migrates m to node b with enough buffer space ($S_b - 1 \geq T_b$) before the buffer becomes full. b will later replicate or migrate m to other encountered nodes. At the same time, a has enough buffer space to accept new messages. The message migration process is shown in Algorithm 2, when nodes a and b are in contact. When either nodes a or b is overloaded, we do the message migration first.

Algorithm 2 Message Migration (at node a)

```
Nodes $a$ and $b$ first exchange the metadata.
if $S_a < T_b$ and $S_b - 1 \geq T_b$ then
    Node $a$ selects the largest hop-count message to migrate to node $b$.
```

IV. IMPLEMENTATION AND SIMULATION

A. Simulation Methods and Setting

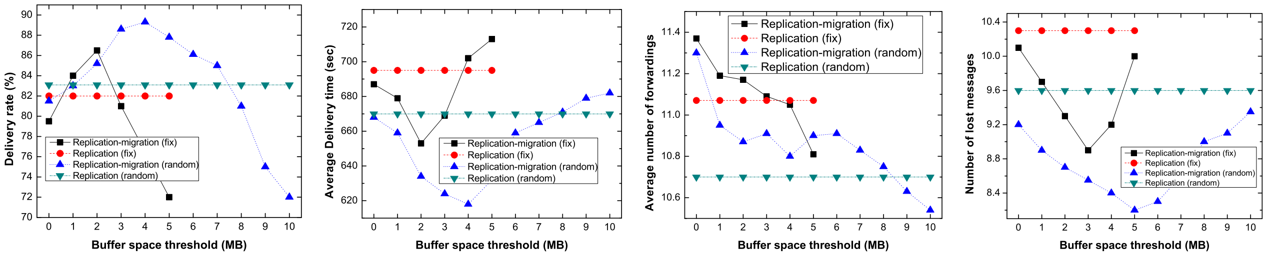
We implement our routing protocol both in synthetic and real mobility traces using MATLAB. *(1) Synthetic trace:* in the synthetic mobility models, we set up a 20-node environment. The mobility pattern of the nodes is followed by the random waypoint model. There are $2,563$ time slots in seconds, and in each time slot, there is a contact between two nodes. *(2) Real trace:* we evaluate our schemes in the Intel trace [14], which includes Bluetooth sightings by groups of users carrying small devices (iMotes) for six days in the Intel Research Cambridge Corporate Laboratory. There is a stationary node and 8 nodes that correspond to mobile iMotes. There are $2,766$ contacts between these nodes over a period of $359,311$ time slots in seconds.

In our simulation, we analyzed three specific scenarios: *Scenario 1*: comparisons of delivery rate, latency, number of forwardings, and number of lost messages between the joint replication-migration scheme and the replication scheme in different buffer space thresholds. *Scenario 2*: comparisons of delivery rate, latency, number of forwardings, and number of lost messages between the joint replication-migration scheme and the replication scheme in different hop-count thresholds. *Scenario 3*: number of forwardings and latency comparisons between using 2-hop neighbor information and just using 1-hop information (its own activity level).

In our simulation, the message size is 1MB. The message generation rate is 1 contact per message. There are a total of 20 messages. In Scenario 1, we evaluate under two conditions: *fixed buffer size* (the buffer size of all nodes is 10MB) and *random buffer size in a range* (each node will assign a random buffer size in a range (20MB)). T_b varies from 0MB to 5MB in the fixed buffer size condition and 0MB to 10MB in the random buffer size condition. In Scenario 2, we assume that the buffer size is infinite. In Scenario 3, the hop-count threshold varies from 0 to 10 in the synthetic trace and 0 to 6 in the real trace. In our simulation, we find that by using a different c in Eq. 1, the results have the same trend. Hence, we set c to 0.5 in this paper.

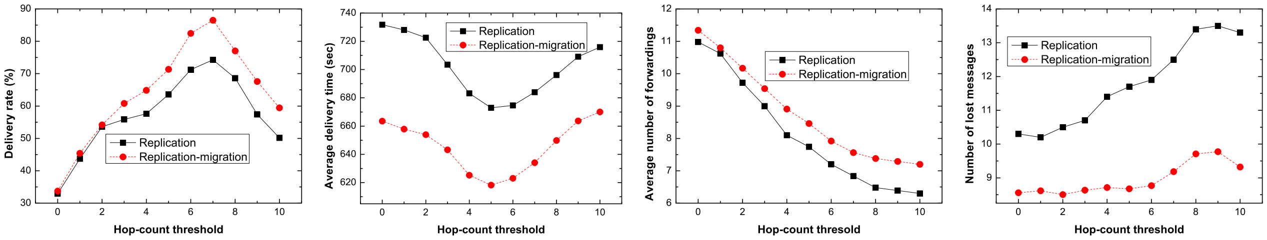
B. Results in Synthetic Trace

Scenario 1: in both the fixed and random buffer size conditions, we find that in an appropriate buffer space threshold, the joint replication-migration scheme can increase the delivery rate and decrease the latency compared to only using the



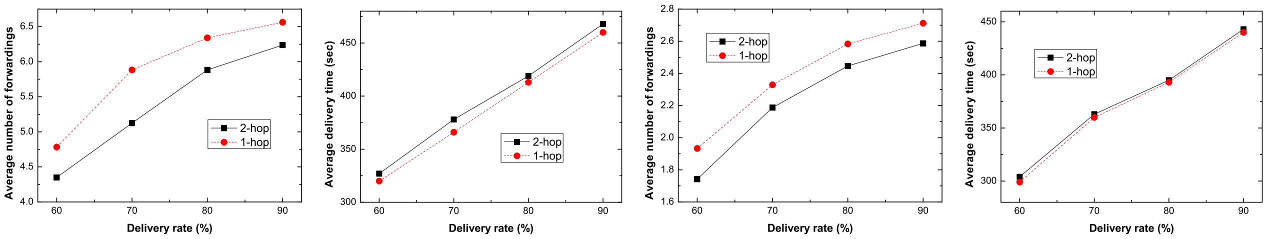
(a) Delivery rate (b) Latency (c) Number of forwardings (d) Number of lost messages

Fig. 2. Comparison between migration and dropping in different buffer space thresholds in the synthetic trace.



(a) Delivery rate (b) Latency (c) Number of forwardings (d) Number of lost messages

Fig. 3. Comparison of delivery rate, latency, number of forwardings, and number of lost messages in different hop-count thresholds in the synthetic trace.



(a) Number of forwardings (b) Latency (c) Number of forwardings (d) Latency

Fig. 4. Comparison between 2-hop and 1-hop in the synthetic trace((a) and (b)) and the real trace ((c) and (d)).

message replication policy in Fig. 2. In Fig. 2(a), we find that in the fixed buffer size condition, the joint replication-migration scheme increases by the most compared to the replication scheme when the buffer space threshold is 2 in delivery rate. In the random buffer size condition, the joint replication-migration scheme performs better than in the fixed condition, which can have a 7.5% delivery rate that increases when the buffer space threshold is 4. The joint replication-migration scheme can dramatically reduce the number of lost messages in all buffer space thresholds in both conditions.

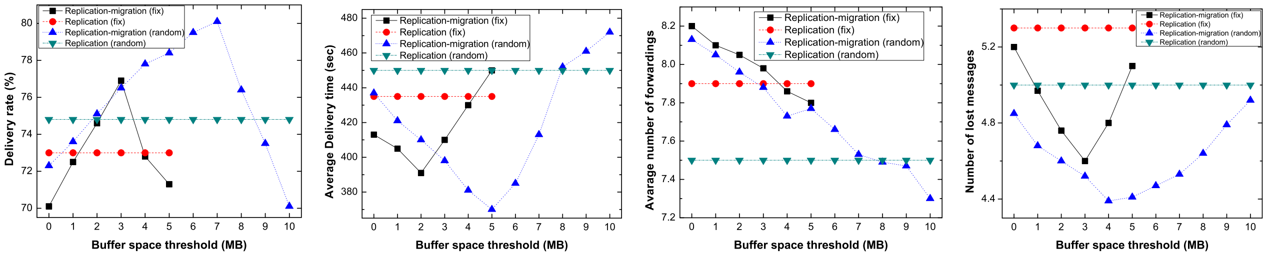
Scenario 2: from Fig. 3(a), we find that the joint replication-migration policy has higher delivery rates compared to the replication policy only. When the hop-count threshold is 5, the improvement is best at about 20%. The joint replication-migration scheme reduces the latency overall by about 5%, as shown in Fig. 3(b). When the threshold is 0, the decreased ratio is about 6.5%, which is the highest one. The lowest one occurs when the threshold is equal to 9. The number of forwardings reduces when the hop-count threshold increases. Using the joint replication-migration policy increases the number of forwardings by about 9% compared with the replication scheme in Fig. 3(c). By using the joint replication-migration policy, there are fewer messages lost in Fig. 3(d).

Scenario 3: from Figs. 4(a) and 4(b), we can see that using 2-hop information can reduce the number of forwardings by about 9.5% but can only increase the latency by about 2%. Therefore, using 2-hop information outperforms using 1-hop information in the synthetic trace. From the simulation results, we find that the delivery rate is similar when using 2-hop and 1-hop neighborhood information to calculate the activity level.

C. Results in Real Trace

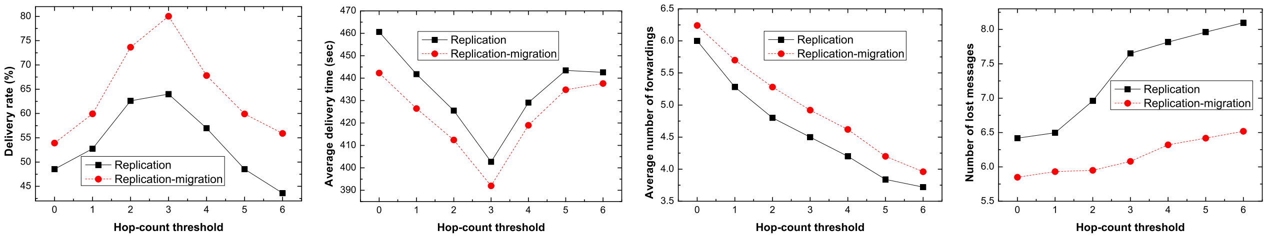
Scenario 1: in the real trace, we find the same results as in the synthetic trace; when choosing an appropriate buffer space threshold, the joint replication-migration scheme can significantly increase the delivery rate and decrease the average delivery time both in the fixed and random buffer size conditions in Fig. 5. In Fig. 5(d), the joint replication-migration scheme can reduce the number of lost messages by about 13.6% in the fixed buffer size condition and by 14.6% in the random buffer size condition.

Scenario 2: As shown in Fig. 6(a), we find that when the hop-count threshold is 3, the joint replication-migration scheme has the best improvement over the replication scheme – about 25%. The joint replication-migration scheme decreases the latency overall by about 4% in Fig. 6(b). When the threshold is 0, the decreased ratio is about 6%, which is the highest one. The lowest one happens when the threshold is equal to 6. The number of forwardings reduces when the hop-count threshold increases, and the joint replication-migration policy increases the number of forwardings slightly compared to using replication only, as shown in Fig. 6(c). From Fig. 6(d), we have the same conclusion as in the synthetic trace. When the hop-count threshold is larger, using the joint replication-migration policy results in a much better performance.



(a) Delivery rate (b) Latency (c) Number of forwardings (d) Number of lost messages

Fig. 5. Comparison between migration and dropping in different buffer space thresholds in the real trace.



(a) Delivery rate (b) Latency (c) Number of forwardings (d) Number of lost messages

Fig. 6. Comparison of delivery rate, latency, number of forwardings, and number of lost messages in different hop-count thresholds in the real trace.

Scenario 3: in Figs. 4(c) and 4(d), we compare the number of forwardings and latency between using 2-hop and 1-hop neighborhood information. We find that by using 2-hop information, the average number of forwardings can be reduced by about 7%, as is seen in Fig. 4(c). At the same time, these two methods have a similar latency.

D. Summery of Simulation

From the simulation results, we can see that using the joint replication-migration scheme outperforms using only the replication scheme when buffer congestion happens in various conditions. In an appropriate buffer space threshold, the joint replication-migration scheme can significantly increase the delivery rate and decrease the average delivery time and the number of lost messages in different conditions. In the random buffer size condition, if the inactive nodes have a larger amount of buffer space to contain the migration messages from the active nodes, which can improve the performance compared with the fixed buffer size condition, the buffer of the inactive nodes will be overloaded by the migration messages quickly. By comparing different hop-count thresholds, we find that when the threshold is too small, the message replication is based on the node selection, and the delivery rate decreases. When the threshold is too large, the message will replicate at every contact; thus, there will be massive copies of messages in the network. Because of the limitations in buffer space and bandwidth, the performance degrades because of the unnecessary copies of messages. Choosing different hop-count thresholds will also degrade the performance of the routing process. Using 2-hop information to calculate the activity level results in a better performance compared to just using 1-hop information.

V. CONCLUSION

In this paper, we proposed an effective routing protocol for DTNs. The message hop-count was used to prioritize the messages stored in the buffer. We introduced the hop-count threshold to control the message replication speed. We used 2-hop information to predict nodes' activity levels, which is a metric that selects a good relay node. To overcome the limited buffer size and bandwidth, we proposed the message migration policy to control buffer congestion. At the same time, we considered the available buffer space threshold and performed evaluations under the fixed and random buffer size conditions. The joint replication-migration-based DTN routing scheme performs well in both synthetic and real traces.

REFERENCES

[1] A. Pentland, R. Fletcher, and A. Hasson, "Daknet: rethinking connectivity in developing nations," *Computer*, vol. 37, no. 1, pp. 78–83, 2004.

[2] J. Wu and Y. Wang, "Social feature-based multi-path routing in delay tolerant networks," in *Proc. of IEEE Infocom*, 2012.

[3] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "Maxprop: Routing for vehicle-based disruption-tolerant networks," in *Proc. of IEEE INFOCOM*, 2006.

[4] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," *Technical Report, Dept. of Computer Science, Duke University*, 2000.

[5] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *Proc. of ACM WDTN*, pp. 252–259, 2005.

[6] J. Wu and Y. Wang, "A non-replication multicasting scheme in delay tolerant networks," in *Proc. of IEEE MASS*, pp. 89 –98, 2010.

[7] Y. Wang, X. Li, and J. Wu, "Multicasting in delay tolerant networks: delegation forwarding," in *Proc. of IEEE GLOBECOM*, 2010.

[8] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Impact of human mobility on opportunistic forwarding algorithms," *IEEE Transactions on Mobile Computing*, vol. 6, pp. 606–620, 2007.

[9] X. Zhang, G. Neglia, J. Kurose, and D. Towsley, "Performance modeling of epidemic routing," *Comput. Netw.*, vol. 51, pp. 2867–2891, July 2007.

[10] A. Krifa, C. Baraka, and T. Spyropoulos, "Optimal buffer management policies for delay tolerant networks," in *Proc. of IEEE SECON*, pp. 260 –268, 2008.

[11] M. Seligman, K. Fall, and P. Mundur, "Alternative custodians for congestion control in delay tolerant networks," in *Proc. of ACM CHANTS*, pp. 229–236, 2006.

[12] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility," in *Proc. of IEEE PERCOMW*, 2007.

[13] T. Hossmann, T. Spyropoulos, and F. Legendre, "Putting contacts into context: Mobility modeling beyond inter-contact times," in *Proc. of ACM MobiHoc*, 2011.

[14] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, "CRAWDAD trace cambridge/haggle/imote/cambridge (v. 2006-01-31)," Jan. 2006.