Timely Information Dissemination with Distributed Storage in Delay Tolerant Mobile Sensor Networks

Aashish Dhungana
Department of Computer Science
Virginia Commonwealth University
401 West Main St. Richmond, VA 23284
dhunganaa@vcu.edu

Eyuphan Bulut
Department of Computer Science
Virginia Commonwealth University
401 West Main St. Richmond, VA 23284
ebulut@vcu.edu

***Abstract*—In delay tolerant mobile sensor networks, the application may sometimes require informing all the sensor nodes in the network about the presence of a message while delivering it to a specific destination node. But, due to constraints on buffer space, it is not efficient to keep these message copies in the buffer of all nodes. In this paper, we propose a probabilistic approach for information dissemination to every node in the network but caching the message copies in only some of the nodes' buffer that can be utilized for future references to decrease the data access latency. This method saves the buffer space of nodes which do not hold the copies while still allowing the whole network nodes to access information on the message whenever required by providing the copies from the nodes that still hold it.**

***Index Terms*—Information dissemination, epidemic routing, buffer management, data access latency, caching.**

I. Introduction

Delay tolerant mobile sensor networks consist of mobile devices with sensors and that connect intermittently. Thus, in such a challenging network environment, data transmission occurs only when two nodes encounter each other. Such connections are intermittent and there is no end-to-end path between nodes making data routing challenging in such scenarios [1], [2]. Specially under extreme conditions like disaster recovery where different sensors have been equipped around an area to collect data from the environment, message transmission and data access becomes more difficult and uncertain. To overcome such challenges in the network, a remarkable amount of studies [3]–[12] have been done and various message routing protocols have been proposed.

In this paper, we consider a relevant but a slightly different scenario in delay tolerant mobile sensor networks. Sometimes it is necessary to send the information collected from one sensor node to another sensor node while informing all other sensor nodes in the network. But due to limitations on buffer space and energy of such networks, it is not efficient to keep the message copies in all nodes' buffer, especially for the applications with large message sizes (e.g., finely collected sensor data or multimedia files). For example, in a military based application, a soldier in the field can share his scan information of the local area (obtained by 3D or thermal cameras) with his commander, but he can also inform other soldiers about this message (i.e., information of a specific location with coordinates) so that once they need this information (when they move and come to the same location later and want to check previous scanned data), they can easily reach it from others who have it. By this way, they do not need to save the data until they really need it. A similar situation can also occur in a crowd-sourced based data collection process in a disaster recovery scenario in which the users may need to see all other users' critical data (e.g., photos/videos) but may not store them all.

Routing of the messages to a destination node in networks with DTN characteristics have been studied extensively. However, in cases where the same data needs to be used by some other nodes, re-routing the same messages again from the source to that destination is both energy-inefficient and time consuming making the data access in such networks very inefficient.

In this paper, we propose a probabilistic approach to deliver a message to its destination while informing all (or most) of the nodes in the network but caching the information in only some of the nodes. This also saves the space in buffers of nodes that do not hold the copy while reducing the data access latency whenever some other nodes in the network have to reach the same information. Our study makes use of epidemic routing because this routing method has the optimal time for forwarding the message to all nodes and our probabilistic approach makes it more energy-efficient by deleting the unnecessary copies and caching it on only few of the nodes in the network. The main goals of our study can be summarized as (i) timely delivery of the message to its intended destination, (ii) informing all the nodes in the network about the message content, (iii) reducing the average buffer load in the network and cache the message copies at a minimum number of nodes based on application needs, and (iv) reducing the data access time for other nodes after the message has been introduced in the network.

We analyze two probability variables that we introduce into our model. Forwarding probability p controls the message forwarding behavior of the nodes and determines the average delay for the message to be delivered and to reach the other

This work was supported in part by NSF award CNS-1647217.

nodes in the network. Similarly, q variable controls the number of copies in the network and helps achieve the required number of caching in the network. We use these two variables only when the node carrying the message meets another node and the usage of q value depends on p. That is, q is only applied if a message is forwarded to another node with probability p. This is to ensure that the message visits every node in the network and hence always gets delivered. However, we also study the cases where q is applied independent of p. In such cases, we target scenarios in which the target reach of the message does not need to be all nodes in the network. The rest of the paper is structured as follows. We discuss the related work in Section II. In Section III, we provide the description of our proposed system. Then, in Section IV, we provide the details of simulation environment we use to carry out experiments and present simulation results. Finally, we conclude the paper and provide future works in Section V.

II. Related Work

As the epidemic routing generates lots of messaging overhead, there have been many routing algorithms proposed to mitigate this problem while keeping the delivery ratio as high as possible. Different techniques (e.g., multi-copy based [1], [2], [13], [14], single-copy based [3]–[5], social relations aware [6]–[9] and erasure coding based [10]–[12]) have been applied to achieve the maximum benefit in different scenarios. In all these works, however, the main focus has always been increasing the delivery ratio of packets at the destination.

In some works [15], vaccination of distributed copies at the nodes in the network have been studied. That is, once the message is delivered, the other nodes carrying a message copy needs to be acknowledged about the delivery of the message so that they can stop carrying it and save space in their buffers. Similarly, some studies [16] use TTL to stop spreading of copies after some hops.

There are also a few previous work that study self-stopping mechanisms for epidemic like algorithms [17]–[20]. These are the closest works to our study in the literature. However, their focus is to distribute the copy of the message to a target reach (i.e., target percentage of nodes in the network) and make the spreading stop by that time. In this paper, we study a more comprehensive solution. While we aim to inform all the nodes or a percent of nodes in the network about the presence of the message, the actual message is preserved only in some nodes in the network. This is to ensure that buffer sizes of nodes are efficiently used while letting each node who may need the message later access easily from the nodes holding the message.

III. Proposed System

We assume that there are M nodes moving on a 2D torus according to a random mobility model. All nodes are identical and have a same transmission range. The meeting times of nodes are assumed to be independent and identically distributed (IID) exponential random variables. The expected inter meeting time between pairs of nodes is denoted as EM.

In order to route a message to its destination while also informing each node in the network about the presence of the message, we base our model on epidemic routing to utilize its optimal delay advantage over other routing algorithms. We introduce two probability variables: p, a forwarding probability and q, a deleting probability such that on the stopping of the routing algorithm, we don't overload the network with message copies and store copies in few of the nodes that can be accessed by other nodes later. When two nodes meet (i.e., come to the communication range of each other), a message is forwarded with probability p. If the message is forwarded, then we cache the copy in the forwarder node's buffer with probability $1 - q$ (i.e., we delete it with probability q). Each message has a unique message id. While a message is routed towards its destination over other nodes in the network, the nodes that don't hold the message copy are also informed about the presence of message and its id. When a node without copy meets any of the other nodes with a copy, the node without the copy requests the messages missing in its buffer. Once the first node forwards its cached copy to the requesting node, the forwarder node deletes its copy with probability q. The receiver node can now act as a new relay node and provide the message to other nodes on request. This ensures that while providing faster data access, we never overload the network with too many message copies. However, a sufficient number of copies can still be hold at some nodes in the network at the end of the process so that the nodes not holding the message copy but are informed about its delivery can easily request to access it when they need it later.

Next, we study the impact of p, and q on the dynamics of routing both individually and jointly. Specifically, we study the impacts on the following performance metrics:

- Average delivery ratio: The ratio of the total number of messages delivered to its intended destination.
- Average delivery delay: The average time passes for the messages since their generation at source nodes until the delivery at their destinations.
- Total coverage: The percentage of nodes informed about the message. Note that not all nodes keep the copy of the message, however, they can reach it from other nodes holding a copy later.
- Average coverage delay: The delay of informing all reachable nodes about the message. In order words, it is time recorded at the copy of the message to the last node.
- Total caching nodes: The percentage of nodes holding a message copy in their buffer.
- Data access latency: The average time required for a node without copy to access a message copy later from the nodes with a cached copy.

A. Probabilistic Timely Message Forwarding with p

In our system model, p represents the message forwarding probability. This is used to control the forwarding behavior of nodes. Lower value of p decreases the chance of message being forwarded, hence increases the message delivery delay.

It might be desirable to have higher values of p, but depending on the delay requirement of the specific application and network parameters, this value can be adjusted. Moreover, although higher values of p significantly reduces the delay, it also exhausts the network resources (i.e., buffer space, energy) very fast which might not be desirable in every application. Proper tuning of this variable can achieve more stable and efficient message routing while providing savings in buffer space to be used in the delivery of high-priority messages.

Each node has some forwarding probability (p) whose value changes in range [0,1]. It can easily be seen that the average time it takes for a node to copy its message to another node with probability p is given by:

$$\begin{aligned}\Delta p &= \sum_{i=1}^{\infty} \frac{EM}{M-1} p(1-p)^{i-1} i \\ &= \frac{EM}{(M-1)p}\end{aligned}$$

Then, using the models in previous studies [1], [2], the average message delivery delay, $\delta_d(p)$, (when $q = 0$) can be estimated using the equation:

$$\delta_d(p) = \frac{EM * H_{M-1}}{(M-1)p}$$

where H_{M-1} is the harmonic number.

The average coverage delay can also be derived as:

$$\begin{aligned}\delta_c(p) &= \sum_{i=1}^{M-1} \frac{EM}{p(M-i)i} \\ &= \frac{EM}{pM} \sum_{i=1}^{M-1} \left(\frac{1}{i} + \frac{1}{M-i}\right) \\ &= \frac{2EM * H_{M-1}}{pM}\end{aligned}$$

Note that the coverage delay is almost double of the delivery delay. This is expected because the destination is selected uniformly and randomly and could be any node between the first node covered and last one covered (this on the average becomes like covering middle node with half of coverage delay).

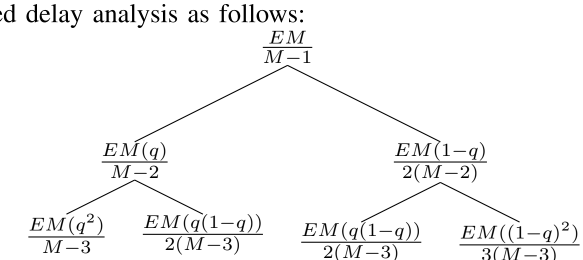
B. Probabilistic Buffer Space Management with q

Our main goal for this study is to provide message information to every node in the network, but keep the message in few nodes' buffer for future reference. Forwarding message with p achieves the earlier goal but to maintain limited number of messages in the cache, we need to delete the extra copies from the network. Every time when a node forwards its message to its neighboring node with probability p, message is kept in the forwarder's buffer with probability $1-q$ (i.e., the message copy from the source node's buffer is deleted with probability q). Similar to the implementation of p, q value has a range of [0,1]. Once the message copy is deleted at the forwarder node, the forwarder node cannot further take part in routing process. These nodes are considered inactive until a stable network state is reached. Once the information is disseminated across the network, this limitation from inactive nodes are removed so that these nodes can also access the data from the nodes holding the copy. Since the copy deletion is instantiated only after message forwarding, it ensures that it doesn't affect the delivery ratio and total coverage achieved from p. However, with higher values of q, delay is slightly increased since the total nodes involved in information dissemination is reduced. It is easy to see that the total number of copies in network for any q can be given by:

$$N = (1-q)M \tag{1}$$

where N is total number of nodes holding the copy (i.e., caching nodes). This formula does not include p as the copy count is mainly determined by q value. For example, with q = 0.9, total nodes that will be holding the message copy in their buffers will be 10% of all nodes, for any p. The system can be modeled accordingly with different values of q.

The average delivery delay of a message with application of q (with p=1) can be calculated using a probabilistic tree based delay analysis as follows:



The source node meets the first node without copy after $EM/(M-1)$ time on the average, and forwards it (with p=1). If this first node met is the destination, then the delay becomes $\delta_{d1}(q)$=$EM/(M-1)$ (i.e. top of the tree). If it is not the destination, then the average time passes to the next meeting of any pairs with a copy opportunity (let's call it eligible pairs) depends on two cases that occur by the deletion decision after the first eligible pair meeting. In the first case, the source node deletes the copy at its buffer with probability q and becomes ineligible to receive it again from other nodes in the future (i.e., left branch in second level). Thus, the time to next eligible pair meeting is $EM/(M-2)$. In the second case, the copy is kept with probability $1-q$, and the time to the next eligible pair meeting is $EM/2(M-2)$ (i.e., right branch in second level). Then, the expected delay for a delivery occurring at second eligible pair meeting becomes:

$$\begin{aligned}\delta_{d2}(q) &= \frac{EM}{M-1} + \left(q\frac{EM}{M-2} + (1-q)\frac{EM}{2(M-2)}\right) \\ &= \frac{EM}{M-1} + \frac{EM}{M-2}\left(\frac{1+q}{2}\right)\end{aligned}$$

Following the same logic, one can easily see that the expected delay for delivery happening at x^{th} eligible pair meeting can be calculated as (by taking the sum of all probabilities at the x^{th} level of the tree):

$$\delta_{di}(q) = EM \sum_{i=1}^{x} \left(\frac{1}{M-i} \sum_{j=1}^{j=x} \frac{q^{j-1}}{i}\right)$$

As the destination could be one of the remaining $M-1$ nodes, then the expected average delivery delay over all cases can be given by:

$$\begin{aligned}\delta_d(q) &= \sum_{x=1}^{M-1} \frac{1}{M-1}\delta_{di}(q) \\ &= \frac{EM}{M-1}\left(\sum_{i=1}^{i=M-1}\sum_{j=1}^{j=i}\frac{q^{j-1}}{i}\right)\end{aligned}$$

Note that the equation becomes EM when $q = 1$, and becomes equal to $\delta_d(p)$ formula with $p = 1$, when $q = 0$. This equation can also be written in terms of harmonic numbers as:

$$\delta_d(q) = \frac{EM}{M-1}\left(\sum_{i=1}^{M-1} q^{i-1}(H_{M-1} - H_{i-1})\right)$$

C. Joint Probabilistic Message Forwarding and Deletion

In previous sections, we analyzed the impact of p and q separately. In this part, we look at how the information dissemination performance is affected when they applied jointly.

We consider two cases in the joint implementation of p and q variables: (i) when deletion decision is given after a successful forwarding (ii) when deletion decision is given independently of forwarding.

In the first case, the delivery ratio always reaches 100% as there will be always a copy in the network. This might be desired for applications where the message query needs to reach all nodes. In such a scenario, the delivery delay can be estimated simply with the following equation:

$$\delta_d(p,q) = \frac{\delta_d(q)}{p}$$

In the second case, the delivery ratio may not always reach to 100%, as the copies available at nodes can be deleted before they are distributed to other nodes. This might be desired for applications where message query needs to reach only a percentage of all nodes. For example, it might not be feasible to collect packets from all of the sensors to achieve the energy-efficiency [21]. This may require sending a command about sampling, sensing, and reporting an event to a certain percentage (e.g., 80%) of sensors in the network. In such a scenario, the number of copies in the network may also be very low depending on p and q relation. The delivery delay will also show different patterns than in dependent case. In simulations, we discuss interesting results.

To derive the number of copy counts in the network when the dissemination stops, let $C(i)$ denote the copy count at the i^{th} eligible pair meeting. Then, we have the following set of equations for $0 < i \leq M-1$ (with $C(0) = 1$):

$$\begin{aligned}C(i+1) &= p(1-q)(C(i)+1) + pq(C(i)) + \\ &\quad (1-p)q(T(i)-1) + (1-p(1-q)(T(i)) \\ &= C(i) + p - q\end{aligned}$$

This shows that at every eligible pair meeting, the copy count increases by $p - q$ on the average. Then, if the R_{max} is the maximum number of nodes that could be reachable (which is also equal to average delivery ratio) in the given (p,q) scenario, then, the eligible pair meeting count becomes R_{max}/p and the copy count saturates at $R_{max}(1 - q/p)$.

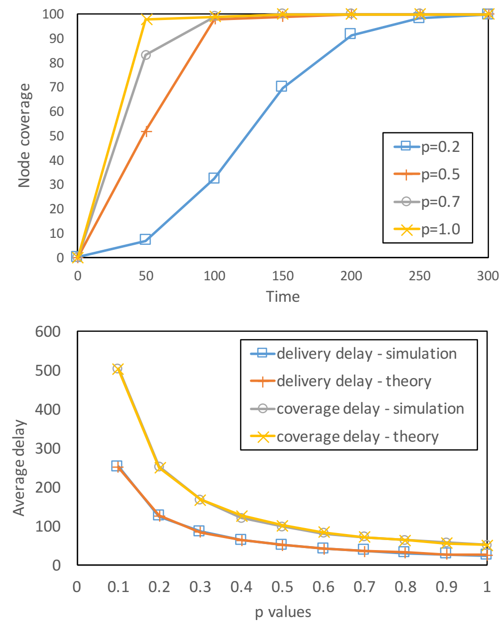


Fig. 1: Results with different p values (with q=0) (a) Delivery delay by time (b) Comparison of theoretical and analysis results for average delivery delay and coverage.

IV. Simulations

In order to evaluate the performance of the proposed dissemination algorithm, we have built a custom simulator using java. We consider Random Walk (RW) model in our simulations as this model is commonly used when studying the DTN routing protocols. Our model is designed using a 300 by 300 torus where all nodes move randomly with the speed of [4-13] m/s for a period of epoch [8-15]s. After an epoch period is reached, the node again randomly chooses its direction, speed and epoch duration and continue with the movement. On reaching the endpoint of torus, the node again starts from the opposite end point of the torus. Under such conditions, it is found that the inter meeting times for the pairs of nodes are exponential and has a mean of 480s [3]. We will be using this fact for the remainder of this paper. It is also assumed that all messages can be transferred as soon as the connection is made and no drops occur due to buffer space. The transmission range for all nodes is taken to be constant of 10m and only one forwarding can be done by a node at a time. In our simulations, we have considered the number of nodes (M) to be 100. All the results obtained from the simulations is the average of 1000 runs.

To implement the probabilistic forwarding and deletion, we generate two random numbers between 0-1 when a connection is established between two nodes. If the node's forwarding probability is greater than the first random number generated,

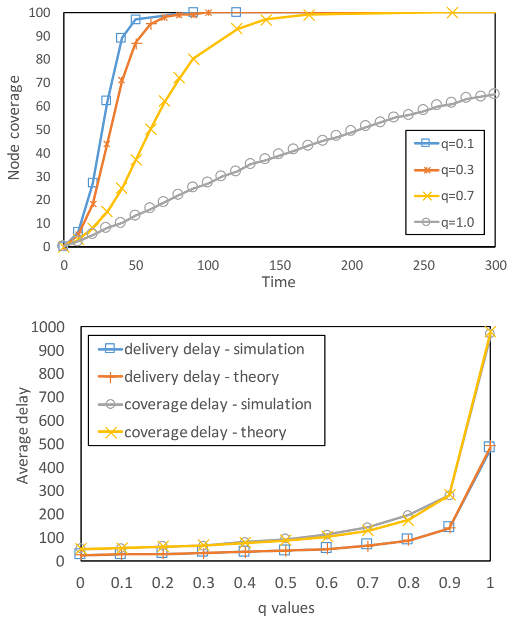


Fig. 2: Results with different q values (with p=1) (a) Delivery delay by time (b) Comparison of theoretical and analysis results for average delivery delay and coverage.

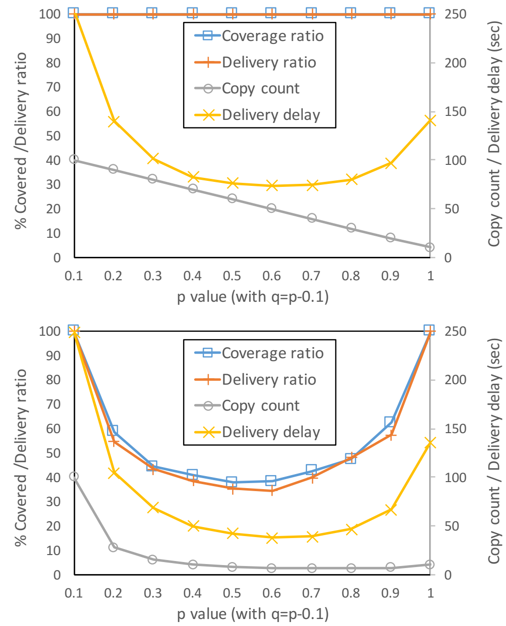


Fig. 3: (p,q) results with q=p-0.1: (a) dependent case (b) independent case.

the message is copied to the new node. Similarly, if the node's deletion probability is greater than the second random number generated, the message is deleted at the forwarder node. In dependent case, deletion occurs only if the message is forwarded.

In Figure 1, results with different p values are shown when q=0. We can see that the delivery delay and coverage delay decreases as p increases. It is also important to note that, whatever the delay is, all the messages are delivered as it is expected per the nature of dependent case definition. From the second figure, we see that coverage delay is approximately double of delivery delay and our theoretical model matches with simulation results.

Similarly, Figure 2 illustrates the results with different q values when p=1. We see that the delay for both delivery and coverage increases as q increases but ultimately, both delay and coverage will reach 100% as expected. The second figure shows coverage delay is approximately double of delivery delay and our theoretical model matches with simulation results.

Next, in Figure 3 we look at different parameters with different p values and $q = p - 0.1$ with the graphs for dependent and independent cases, respectively. We see that the coverage and delivery ratio are not always 100% in case of independent p and q which is indeed expected. Due to the fact that deletion decision is made without the consideration of forwarding decision, the copy count in the second case decreases too rapidly for small value of p whose impact can as well be seen in the delivery ratio of the network. Also the delay is smaller in the second case as it shows only the average delay

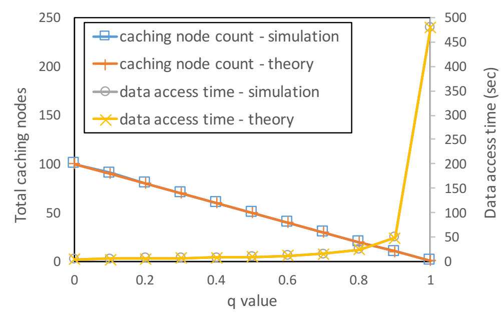


Fig. 4: Number of caching nodes in dependent case and data access delay.

of delivered messages which is not 100%. In Figure 4, the data access time from the caching nodes together with different caching node counts for different q values in dependent case are given. Our simulation results are consistent with theoretical results. We can see that data access time reduces as the number of copies increases.

In Figure 5, we show the percentage of covered nodes for all (p,q) pairs in independent case. In other words, deletion by q is applied independently of forwarding decided by p. As the q value increases or p decreases coverage ratio decreases. Note that these results are also equal to the maximum delivery ratios reachable for given (p,q) pairs. In other words, in applications in which notifying or sampling from a certain percentage of nodes is sufficient, application of q can be considered independent of application of p.

Finally in Figure 6, we show the contour plot of the copy counts stay in the network (i.e., caching nodes) by the time

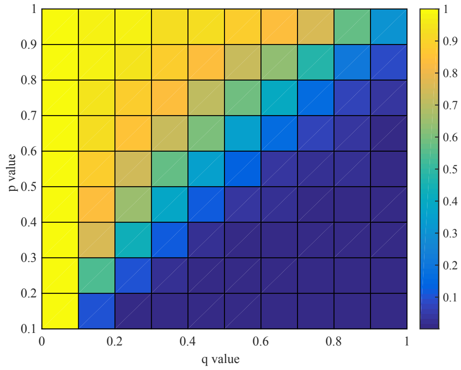


Fig. 5: Percentage of covered nodes for all (p,q) pairs in independent case.

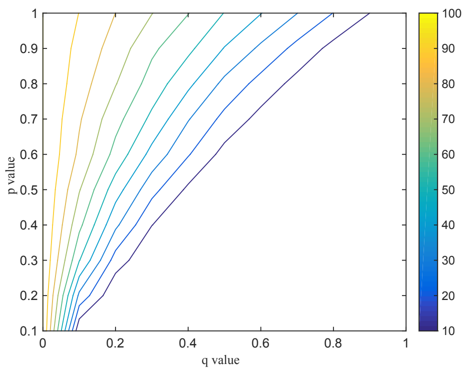


Fig. 6: Caching node (copy) counts in the network for all (p,q) pairs in independent case.

dissemination stops. First of all, we verified these values using the $R_{max}(1-q/p)$ formula, where R_{max} values are presented in Figure 5. As p increases and the q decreases, the copy count increases. However, once q is equal to or bigger than p, the value becomes zero, as expected.

V. Conclusion

In this paper, we study an interesting problem of joint information dissemination and distributed storage of that information at limited number of nodes for future reference. Utilizing probabilistic approaches, we aim to control the forwarding and deletion of message copies at the nodes. Depending on the application of these probabilities, we derive the delivery delay and copy counts analytically and verify with simulations. In the future work, we will obtain more results and derive all other metrics analytically. We will also perform simulations with real DTN traces for which we will use dynamically tuned p and q values for every node depending on their relations with other nodes.

References

[1] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: the multiple-copy case," *IEEE/ACM transactions on networking*, vol. 16, no. 1, pp. 77–90, 2008.

[2] E. Bulut, Z. Wang, and B. K. Szymanski, "Cost-effective multiperiod spraying for routing in delay-tolerant networks," *IEEE/ACM Transactions on Networking (ToN)*, vol. 18, no. 5, pp. 1530–1543, 2010.

[3] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: The single-copy case," *IEEE/ACM Transactions on Networking (ToN)*, vol. 16, no. 1, pp. 63–76, 2008.

[4] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "Maxprop: Routing for vehicle-based disruption-tolerant networks." in *INFOCOM*, vol. 6, 2006, pp. 1–11.

[5] E. Bulut, S. C. Geyik, and B. K. Szymanski, "Utilizing correlated node mobility for efficient DTN routing," *Pervasive and Mobile Computing*, vol. 13, pp. 150–163, 2014.

[6] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: Social-based forwarding in delay-tolerant networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 11, pp. 1576–1589, 2011.

[7] E. Bulut and B. K. Szymanski, "Exploiting friendship relations for efficient routing in mobile social networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 12, pp. 2254–2265, 2012.

[8] E. M. Daly and M. Haahr, "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*. ACM, 2007, pp. 32–40.

[9] E. Bulut, Z. Wang, and B. K. Szymanski, "Impact of social networks on delay tolerant routing," in *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE.* IEEE, 2009, pp. 1–6.

[10] Y. Wang, S. Jain, M. Martonosi, and K. Fall, "Erasure-coding based routing for opportunistic networks," in *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*. ACM, 2005, pp. 229–236.

[11] E. Bulut, Z. Wang, and B. K. Szymanski, "Cost efficient erasure coding based routing in delay tolerant networks," in *Communications (ICC), 2010 IEEE International Conference on*. IEEE, 2010, pp. 1–5.

[12] Y. Wang and H. Wu, "Delay/fault-tolerant mobile sensor network (dft-msn): A new paradigm for pervasive information gathering," *IEEE Transactions on mobile computing*, vol. 6, no. 9, pp. 1021–1034, 2007.

[13] E. Bulut, Z. Wang, and B. Szymanski, "Time dependent message spraying for routing in intermittently connected networks," in *IEEE GLOBECOM 2008-2008 IEEE Global Telecommunications Conference*. IEEE, 2008, pp. 1–6.

[14] E. Bulut and B. K. Szymanski, "Secure multi-copy routing in compromised delay tolerant networks," *Wireless personal communications*, vol. 73, no. 1, pp. 149–168, 2013.

[15] T. Matsuda and T. Takine, "(p, q)-epidemic routing for sparsely populated mobile ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 5, pp. 783–793, 2008.

[16] A. Al Hanbali, P. Nain, and E. Altman, "Performance of ad hoc networks with two-hop relay routing and limited packet lifetime (extended version)," *Performance Evaluation*, vol. 65, no. 6, pp. 463–483, 2008.

[17] Z. Chen and C. Chen, "Self-stopping epidemic routing in cooperative wireless mobile sensor networks," in *Wireless and Mobile Computing, Networking and Communications (WiMob), 2016 IEEE 12th International Conference on*. IEEE, 2016, pp. 1–7.

[18] J. Liu and N. Kato, "A markovian analysis for explicit probabilistic stopping-based information propagation in postdisaster ad hoc mobile networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 1, pp. 81–90, 2016.

[19] Y. M. Ko and N. Gautam, "Epidemic-based information dissemination in wireless mobile sensor networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 18, no. 6, pp. 1738–1751, 2010.

[20] R. Vogt, J. Aycock, and M. J. Jacobson, "Quorum sensing and self-stopping worms," in *Proceedings of the 2007 ACM workshop on Recurring malcode*. ACM, 2007, pp. 16–22.

[21] S. Bandyopadhyay, Q. Tian, and E. J. Coyle, "Spatio-temporal sampling rates and energy efficiency in wireless sensor networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 13, no. 6, pp. 1339–1352, 2005.