Optimizing Message Ferry Scheduling in a DTN

Sankardas Roy
Bowling Green State University
Department of Computer Science
sanroy@bgsu.edu

Daniele Tomasi
University of Padua
Department of Mathematics
morris@morrisda.com

Mauro Conti*
University of Padua
Department of Mathematics
conti@math.unipd.it

Shiva Bhusal
Bowling Green State University
Department of Computer Science
sbhusal@bgsu.edu

Arkajyoti Roy
University of Texas at San Antonio
Department of MSS
arkajyoti.roy@utsa.edu

Jiang Li
Howard University
Department of EECS
jli@howard.edu

ABSTRACT

We consider a special type of Delay Tolerant Network (DTN), called "Local-Ferry-based network" (LFN), which enables communication among multiple nodes distributed over a geographic terrain. LFN utilizes controllable special purpose vehicles called "pigeons" to transfer messages among neighboring nodes: some (or all) nodes own these message ferries (*a.k.a.* pigeons), which help in establishing communication links among local nodes, cumulatively setting up the whole network. One research challenge is to schedule the pigeon movement between local nodes (*i.e.*, deciding the pigeon's visiting sequence of the local nodes) to achieve good networking performance (*e.g.*, message delay). Solving this research challenge poses promise for many exciting applications, such as using drones to enable communication among segregated regions in disaster recovery, to augment/connect in-situ IoT deployments, and more. In this paper, we address the above challenge whereas we contribute the following. First, we analyze what it takes to optimize the scheduling algorithm for a pigeon. Second, using the above results we design multiple variants of scheduling algorithms, and we compare their performance with the theoretical optimal delay and with the state-of-the-art algorithms through simulation experiments. Both theoretical analysis and simulation results show the efficacy of our solution. For instance, our best scheduling algorithm achieves (within 5% margin) the theoretical optimal (per-hop) message delay.

CCS CONCEPTS

• **Networks** → Link-layer protocols;

KEYWORDS

DTN, Message Ferry, Scheduling Algorithm, Optimization

*Mauro Conti is also affiliated with the Department of Electrical Engineering, University of Washington, Seattle, USA. This work is partially supported by the EU TagItSmart! Project (agreement H2020-ICT30-2015-688061), the EU-India REACH Project (agreement ICI+/2014/342-896), and the grant n. 2017-166478 (3696) from Cisco University Research Program Fund and Silicon Valley Community Foundation.

MobiWac '18, October 28–November 2, 2018, Montréal, Québec, Canada

ACM ISBN 978-1-4503-5962-7/18/10…$15.00
https://doi.org/10.1145/3265863.3265884

ACM Reference Format:
Sankardas Roy, Daniele Tomasi, Mauro Conti, Shiva Bhusal, Arkajyoti Roy, and Jiang Li. 2018. Optimizing Message Ferry Scheduling in a DTN. In *MobiWac '18: 16th ACM International Symposium on Mobility Management and Wireless Access, Oct. 28–Nov. 2, 2018, Montréal, Québec, Canada.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3265863.3265884

1 INTRODUCTION

Delay Tolerant Network (DTN) [3] supports communications in adverse environments (such as natural disasters) where the damaged infrastructure leads to a partitioned network lacking real-time communication. DTN can also potentially augment/connect in-situ IoT deployments. The assistantship from entities with good mobility is needed in a DTN to set up communication links. These entities can be drones, vehicles, people, or even animals. Prior researchers have explored two mobility modes so far: uncontrolled mobility [11] and controlled mobility [12]. As one example of controlled mobility, an LFN uses special purpose vehicles called *pigeons* [4] to convey messages among local nodes. A *pigeon* is like a message ferry [12] except that it is *owned* by a particular host node and it facilitates message transfer between the owner node (called *home host*) and other local nodes (called *foreign hosts*). The movement of a pigeon is confined within the neighboring region of the home host. Assuming there exist multiple nodes in the network, which own a pigeon each, the above design leads to a special type of DTN involving multiple message ferries controlled on a regional basis. This is aligned with the philosophy of "divide and conquer" paradigm. One example LFN is illustrated in Figure 1 , which has three nodes owning one pigeon each whereas a pigeon transfers messages among local nodes. Some of the prior works [12] used a "global" message ferry that facilitates communication among all nodes of the network, and such a design faces the scalability challenge and management conflict. The use of localized message ferries in our design of LFN is an attempt to address such vexing issues.

In an LFN, a message can traverse from the originating node (potentially) through multiple hops before reaching the final destination node. At each hop, the communication is made possible by a home host and its pigeon. As an example, in Figure 1, there is a path from N_1 to N_5, which has two parts: The first part $N_1 - N_2 - N_4$ is made possible by home host N_2 and its pigeon (realizing links $N_1 - N_2$ and $N_2 - N_4$) whereas the second part $N_4 - N_6 - N_5$ is made possible by home host N_6 and its pigeon (realizing links $N_4 - N_6$ and $N_6 - N_5$). We can draw an analogy between a *home host* node (say H) and a regular router (*e.g.*, on the Internet) whereas H's *pigeon* plays the role of the communication link. Like regular networks (*e.g.*, the Internet), an LFN uses a routing algorithm, which enables each node to find the best next hop in routing a message while the hop-to-hop communication is guided by a localized scheduling algorithm.

Contribution. The *main contributions* of this paper are as follows: (a) We design a *novel optimization framework* for scheduling a pigeon so that the hop-to-hop message latency (*i.e.*, link layer delay) is reduced. Using these results, we design multiple algorithms (variants) for scheduling the pigeon. (b) Both theoretical analysis and simulation results show the efficacy of our solution. For instance, the best scheduling algorithm of ours achieves (within 5% margin) the theoretical optimal (per-hop) message delay.

2 A LOCAL-FERRY-BASED NETWORK (LFN)

In an LFN, the region within which home host H lets its pigeon P move is known as the *cell* of H. A cell can contain more nodes (in addition to the home host), which are denoted as foreign hosts of P [4]. A pigeon can carry the messages between its home host and foreign hosts. However, the pigeon will not carry any message outside its cell (by definition of cell). Thus, LFN involves multiple message ferries that are controlled on a regional basis.

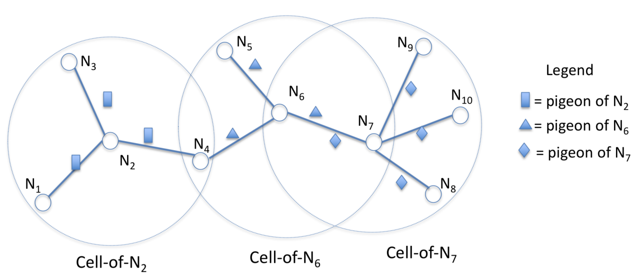

Figure 1: An example LFN consisting of 3 cells

In an LFN, a message can traverse from the originating host node through multiple host nodes before reaching the final destination host. To achieve this, a host node X_1 chooses the best next hop X_2 for a particular message m via a specific routing algorithm. If host X_1 owns a pigeon P_1, it may schedule P_1 to deliver message m to X_2. Host X_2 may similarly do next hop selection, and so on. Technically, any routing protocol can be deployed on LFN. In an example LFN as illustrated in Figure 1, say host N_1 has messages destined for host N_{10}. Host N_1 discovers that host N_2 is the best next hop toward host N_{10}, so it uses host N_2's pigeon (say P_2) to transfer the messages to host N_2. Consecutively, host N_2 forwards the messages using P_2 to its best next hop, host N_4. Now host N_4 discovers that the best next hop for the received messages are host N_6 and it uses pigeon P_6 of host N_6 for this message transfer. This process continues until the messages are delivered to host N_{10}. This example demonstrates that home hosts (*e.g.*, hosts N_2, N_6, and N_7) in the LFN are acting similarly as routers on the Internet.

We observe from the above example that intra-cell traffic (single hop) gives rise to inter-cell traffic (multi-hop). The inter-cell traffic is guided by the routing protocol while the intra-cell traffic is guided by the schedule of the pigeon movement in that particular cell. To address the scheduling problem, we need to study a cell (*i.e.*, the basic unit of LFN) consisting of a home host, the pigeon, and a group of foreign hosts served by this pigeon. As shown in Figure 1, the whole LFN can be viewed as being composed of multiple cells. Note that neighboring cells may overlap as in Figure 1 where cell-of-N_6 and cell-of-N_7 share one link.

Recall that scheduling problem corresponds to a single cell. For the sake of simplicity, we assume that the pigeon travels at a constant speed. When pigeon P is close to a foreign host or home host, P picks up or delivers messages through a high-bandwidth wireless interface. With the advancement of wireless communication, the available short-distance bandwidth grows rapidly and the time spent on exchanging messages are relatively shorter than the travel time. One research challenge is how to obtain *an optimal schedule* for a pigeon to go through all hosts in a cell so that *the delay of messages is minimized.*

3 OPTIMIZING PIGEON SCHEDULING

3.1 Designing the Pigeon Trajectory

Let us recall the construction of a cell in an LFN. A cell has a home host (denoted as H) and n foreign hosts that are denoted as $\{F_1, F_2, \ldots, F_n\}$. For instance, in the leftmost cell in Figure 1, N_2 is the home host and N_1, N_3, and N_4 are the foreign hosts. Home host H owns a pigeon P which carries the burden of passing messages of the cell. A foreign host F_i may have messages to be transferred to H, and H may also have messages to be transferred to F_i, and all of these messages are transferred by pigeon P. Note that P does not directly transfer messages from a foreign host F_i to another foreign host F_j, but such messages can be transferred via an extra hop at home host H. Pigeon P's such home-centric behavior is consistent with the fact that home host owns P and hence P gives priority to home host's communication capability. Note that it is possible that foreign host F_i may also own a pigeon that may set a direct communication link between F_i to F_j. However, that link is not a part of the current cell.

In particular, the message ferrying capability of P emulates a bi-directional communication link between H and each F_i. To do that, P repeatedly visits each F_i as well as H and facilitates in transferring messages in either direction, *i.e.*, from F_i to H and from H to F_i. Now the question is what should be the trajectory of pigeon P to visit the home host and foreign hosts. In an LFN cell, we use the following trajectory for the pigeon: P starts from H, visits a foreign-host F_i, immediately comes back to H, and after that P similarly visits the same or another foreign host. P's such movement may continue again and again, assuming new messages keep on arriving, which are to be transferred to the next hop. Note that P's such movement forms a star-trajectory.

3.2 Formulation of optimized scheduling

In an example of star-trajectory, say there are three foreign hosts in a cell, *i.e.*, F_1, F_2, and F_3, and suppose the pigeon visits them in a round-robin fashion, *e.g.*, pigeon's visiting sequence is as follows:

$$S = << H, F_1, H, F_2, H, F_3, H, F_1, H, F_2, H, F_3, H, \ldots >> \quad (1)$$

We assume that the traffic rate (*i.e.*, message arrival rates) in the cell remains constant (in steady state), and thus the pigeon's visiting sequence of hosts should have a repeating pattern. In the visiting sequence S above, the shortest sub-sequence s which repeats itself to make the whole sequence S is called the visiting cycle. In the above example, $s =< H, F_1, H, F_2, H, F_3 >$ is the visiting cycle. We observe that pigeon's visiting foreign hosts with the same frequency (as in the round-robin fashion above) might be ideal if all foreign hosts have equal impact on the overall message delay. However, if a foreign host F_i's message rate (denoted as λ_i) is much more than other foreign hosts', then it makes sense for P to visit F_i (comparatively) more frequently because that will reduce the overall message delay of the cell. Also, we observe that if a foreign host F_i's distance (denoted as a_i) from home host is larger than others', then visiting F_i makes other foreign hosts wait longer (for the pigeon) and that (potentially) increases the overall message delay. So, the impact of F_i's message rate (*i.e.*, λ_i) and that of F_i's distance (*i.e.*, a_i) are in two opposite directions. In a visiting cycle, the visiting frequencies of the foreign hosts are denoted by $f_1, f_2, \ldots, f_n$, respectively. In the aforementioned round-robin visiting sequence example, each of $f_1, f_2, and f_3$ is equal to 1. To give an example with unequal visiting frequencies, let us consider the following visiting sequence:

$$S' =<< H, F_1, H, F_2, H, F_3, H, F_2, H, F_1, H, F_2, H, F_3, H, F_2, H, \ldots >>, \quad (2)$$

where $s' =< H, F_1, H, F_2, H, F_3, H, F_2 >$ is the visiting cycle, and $f_1 = 1, f_2 = 2, f_3 = 1$, *i.e.*, F_2 is visited with double frequency than other foreign hosts.

We assume that the pigeon has a constant speed, and without loss of generality, we consider that the pigeon moves with the unit speed. So, the pigeon's traversed distance and time taken are synonymous, which allows us to use distance and time interchangeably in our analysis. Our goal is to find $f_1, f_2, \ldots, f_n$ such that the overall message delay of the cell is minimum. More formally, let us define *intra-cell delay of a message* as the time interval between message arrival at H (or at F_i) and delivery to the next hop foreign host F_i (or H). We observe that the intra-cell delay of a message has two parts: *waiting time* that is the interval between time instance of message arrival and the time instance of message being picked up by the pigeon, and *ride time* that is the time the message spends on the pigeon until being delivered to the next hop. Say P takes t_i time to traverse the total distance of H to F_i and F_i to H, *i.e.*, $t_i = 2a_i$. So, the cycle time $T = f_1 t_1 + f_2 t_2 + \ldots + f_n t_n$ where $f_1, f_2, \ldots, f_n$ are foreign hosts' visiting frequencies in the cycle.

Let us focus on P's visits to foreign host F_i. We assume that these visits to foreign host F_i are uniformly distributed (to achieve good networking characteristics, *e.g.*, to minimize jitter) in the cycle. The inter-visit time (denoted by b_i) of a foreign host F_i is the time elapsed between P's two consecutive visits to F_i. For instance, in visiting cycle s' (in the example above), b_1 (*i.e.*, time to traverse the path $F_1 - H - F_2 - H - F_3 - H - F_2 - H - F_1$) is same as T while b_2 (*i.e.*, time to traverse the path $F_2 - H - F_3 - H - F_2$) is half of T because P visits F_2 twice in the cycle. In general, we get $b_i = T/f_i$.

Now let us find the relation between the inter-visit time b_i of foreign host F_i and the average *waiting time* $W_{i,0}$ of messages at F_i (which are destined to H). Note that when pigeon P returns to F_i, pigeon P will pick up all messages in the queue of F_i. For the message at the head of queue at F_i, which arrived soon after P's last visit of F_i, the waiting time can be approximated as b_i. In contrast, the message at the tail of the queue is picked up soon after its generation and the corresponding waiting time is approximately 0. With the constant arrival assumption of messages, the average waiting time before pickup for messages on foreign host F_i is $W_{i,0} = \frac{b_i}{2}$. So, $W_{i,0} = \frac{T}{2f_i}$. On the other hand, the *ride time* ($R_{i,0}$) of such a message is $t_i/2$ because that is the time the message spends on the pigeon (as pigeon moves in a straight line from F_i to H).

Assuming uniform distribution of message arrival at F_i destined to H, the number of messages picked up in one visit is $\lambda_{i,0}.b_i = \frac{\lambda_{i,0}T}{f_i}$ where $\lambda_{i,0}$ is the message arrival rate at F_i, which are destined to H. And, adding the waiting time and ride time, we get the average delay of F_i's messages (to H) is $(\frac{T}{2f_i} + \frac{t_i}{2})$. For messages from the home host H to foreign host F_i, the same conclusion can be reached, *i.e.*, $W_{0,i} = \frac{b_i}{2}$, number of messages picked up in one visit is $\lambda_{0,i}.b_i = \frac{\lambda_{0,i}T}{f_i}$, and the average delay of such messages is $(\frac{T}{2f_i} + \frac{t_i}{2})$. So, for messages corresponding to (from or to) F_i the average delay is $d_i = (\frac{T}{2f_i} + \frac{t_i}{2})$ Note that the total message rate corresponding to (from or to) F_i is $\lambda_i = \lambda_{0,i} + \lambda_{i,0}$. Making weighted average of the above delay over all foreign hosts F_i, $i = 1, 2, \ldots, n$, the overall average intra-cell delay can be computed, which is as follows.

$$\begin{aligned} d &= \frac{1}{(\lambda_1 + \lambda_2 + \ldots + \lambda_n)} \sum_{i=1}^{n} \lambda_i d_i \\ &= \frac{1}{(\lambda_1 + \lambda_2 + \ldots + \lambda_n)} \sum_{i=1}^{n} \lambda_i (\frac{T}{2f_i} + \frac{t_i}{2}) \\ &= \frac{(f_1 t_1 + f_2 t_2 + \ldots + f_n t_n)}{2(\lambda_1 + \lambda_2 + \ldots + \lambda_n)} \left(\frac{\lambda_1}{f_1} + \frac{\lambda_2}{f_2} + \ldots + \frac{\lambda_n}{f_n} \right) \\ &\quad + \frac{1}{2(\lambda_1 + \lambda_2 + \ldots + \lambda_n)} \sum_{i=1}^{n} \lambda_i t_i \end{aligned} \quad (3)$$

We want to compute the visiting frequencies f_i, $i = 1, 2, \ldots, n$ which will minimize d. Note that the second part of Equation 3 is independent of f_i. Therefore, doing partial differentiation and equating partial derivatives to zero, we get the following optimal frequency f_i^* for all $i = 1, 2, \ldots, n$.

$$f_i^* = \sqrt{\frac{\lambda_i (f_1^* t_1 + \ldots + f_{i-1}^* t_{i-1} + f_{i+1}^* t_{i+1} + \ldots + f_n^* t_n)}{t_i \left(\frac{\lambda_1}{f_1^*} + \ldots + \frac{\lambda_{i-1}}{f_{i-1}^*} + \frac{\lambda_{i+1}}{f_{i+1}^*} + \ldots + \frac{\lambda_n}{f_n^*} \right)}} \quad (4)$$

Given the message arrival rates (*i.e.*, λ_i) and the distances a_i (whereas $t_i = 2a_i$), it is not straightforward to compute the optimal visiting frequencies because the expression of f_i^* (ref. Equations 4) depends on f_j^*, where $i, j = 1, 2, \ldots, n$ and $i \neq j$. However, we observe that f_i^* is proportional to square root of λ_i and inversely proportional to the square root of t_i. This observation helps us find out optimal solutions f_i^*, $i = 1, 2, \ldots, n$ as the following. First, we let $f_j^* = k\sqrt{\frac{\lambda_j}{t_j}}$ for $j = 1, 2, \ldots, (i-1), (i+1), \ldots, n$. Then, if we substitute these f_j^* in Equation 4, we get $f_i^* = k\sqrt{\frac{\lambda_i}{t_i}}$. This proves that the above is a feasible solution to Equation 4, *i.e.*, the optimum visiting frequency. Substituting $f_i^* = k\sqrt{\frac{\lambda_i}{t_i}}$, $i = 1, 2, \ldots, n$ in the objective function (Equation 3), we get the optimal overall average delay as follows.

$$d^* = \frac{(\sqrt{\lambda_1 t_1} + \sqrt{\lambda_2 t_2} + \ldots + \sqrt{\lambda_n t_n})^2}{2(\lambda_1 + \lambda_2 + \ldots + \lambda_n)} + \frac{1}{2(\lambda_1 + \lambda_2 + \ldots + \lambda_n)} \sum_{i=1}^{n} \lambda_i t_i \quad (5)$$

Furthermore, only the ratio of the foreign hosts' parameters matters, *i.e.*, as long as the ratio of distances (*i.e.*, $t_i/t_j, \forall i,j$) remain same, and the ratio of arrival rates (*i.e.*, $\lambda_i/\lambda_j, \forall i,j$) remain same, the ratio of optimal frequencies (*i.e.*, $f_i^*/f_j^*, \forall i,j$) are the same.

3.3 The Proposed Scheduling Algorithms

Below we briefly present our two algorithms (*i.e.*, *probabilistic scheduling algorithm*, and *deterministic scheduling algorithm*), which are basically the approximate implementations of the optimal scheduling. Each algorithm has two steps, (i) initialization step and (ii) execution rounds where in each round pigeon P selects and visits one foreign host. Note that the second step continues forever.

3.3.1 Probabilistic scheduling algorithm. Initialization: Pigeon P computes the ratio of ideal visiting frequencies, which follows from the analysis in Section 3.2.

$$f_1^* : f_2^* : \ldots : f_n^* = \sqrt{\frac{\lambda_1}{t_1}} : \sqrt{\frac{\lambda_2}{t_2}} : \ldots : \sqrt{\frac{\lambda_n}{t_n}} \quad (6)$$

Execution round: Pigeon P randomly selects one foreign host F_i with probability p_i where p_i is $\frac{f_i^*}{(f_1^*+f_2^*+\ldots+f_n^*)}$ conforming to Equation 6. If F_i is chosen, then P visits F_i and comes back to H.

3.3.2 Deterministic scheduling algorithm. Initialization: Pigeon P computes a fixed visiting cycle s as follows, which conforms to the theoretical (Equation 6) optimal ratio (of frequencies) as close as possible. (a) The visiting frequencies (*i.e.*, $f_1', f_2', \ldots, f_n'$) of foreign hosts, are computed as $f_i' = k\frac{f_i^*}{f_{min}^*}$ where f_{min}^* is the minimum of all f_i^*s (Equation 6) and k is a positive integer. The value of k is incremented starting from 1 until the ratio of visiting frequencies are close (*i.e.*, within an error bound δ) to what is suggested by optimization analysis (*i.e.*, Equation 6). Basically, we stop at that value of k which makes $|1 - \frac{f_i''}{f_i'}| < \delta$ where f_i'' is the rounded (integer) value of f_i'. (b) We consider that F_i's visiting frequency is f_i'' for each i. Now an auxiliary algorithm is used to compute the visiting cycle s, ensuring that f_i'' visits to foreign host F_i are uniformly spanned for each i.

Execution round: Following the above visiting cycle s, pigeon P chooses the next foreign host F_i to visit. Then, P visits F_i and comes back to H.

3.4 Other Scheduling Algorithms for Comparative Study

As a baseline, one can think of a naive scheduling approach where the pigeon does not attempt to do any optimization and visits the foreign hosts simply in the round robin fashion. Let us call it *round-robin scheduling algorithm.* To the best of our understanding, the state-of-the-art is *Deficit-Round-Robin-based-scheduling* algorithm [7], which is known as DRR. However, we observe that the above DRR algorithm [7] is not readily applicable to the LFN scheduling for pigeon's star trajectory. Note that in original DRR algorithm, pigeon does not return to a particular host (like the home host in our star trajectory) in each round. Hence, we have adapted the original DRR algorithm (while keeping the main idea same) to our scenario to ensure a fair comparison.

3.5 Verification via Simulation

3.5.1 Simulation Setup. We have built a discrete event simulation platform that is able to realize an LFN. We have written the simulation program in JavaScript on top of our own library (written in JavaScript) for discrete event simulation. This choice was motivated by the fact that JavaScript environments (such as the browser) offer APIs for easily rendering in real-time the results of the simulations and to change the configurations while the simulation is running to perform easier analysis. The library allows to easily generate network configurations and to reset the state and to switch the scheduling algorithm of pigeons. To study the performance of a scheduling algorithm, in this experiment setup, we simulate one cell of the LFN. In summary, our program runs on a single thread, leveraging asynchronous execution. The basic scenario we study is a cell, consisting of n foreign hosts uniformly distributed in a 2 unit × 2 unit square area while the home host is located at the center.

3.5.2 Simulation Results. As our optimization analysis suggests, the performance of a scheduling algorithm depends on distribution of the ratio r_i where $r_i = \lambda_i/a_i$, foreign host i's message rate is λ_i, and its distance from home host is a_i. Note that the foreign hosts' ratios being close to each other means the foreign hosts in the cell have similar impact on the intra-cell message delay while the ratios being apart indicates that the foreign hosts have dissimilar impact. To be concrete, let us define an asymmetry factor of a cell, which is $\max(r_i)/\min(r_i)$. Note that if the asymmetry factor of a cell is small, the scheduling task is easier, and even a simple round-robin algorithm works fine (*i.e.*, has low delay). A higher asymmetry factor poses a real challenge to a scheduling algorithm and makes a rigorous performance evaluation possible. So, to cover a wider scope of study, we run two experiments: In the first experiment, the cell configuration is chosen such that the asymmetry factor is small (set to 10) whereas in the second experiment, the asymmetry factor is higher (set to 100). For each experiment, we randomly generated 30 configurations for the cell by varying multiple parameters, *e.g.*, number (n) of foreign hosts in the cell, choosing random locations of the foreign hosts, and varying message rates of the foreign hosts. The value of n is randomly chosen from the range starting from 4 to 30. For each experiment, we compute the 95% confidence interval for the set of the measurements, and unless otherwise mentioned, the confidence interval is less than 5% of the reported result.

The result summary of the first experiment (*i.e.*, involving more-or-less symmetric cells) is reported in Figure 2, which compares the average message delay of all the four scheduling algorithms. It also shows the theoretical optimal delay (Equation 5). We observe that deterministic scheduling gives better performance compared to the probabilistic approach and round-robin scheduling. Furthermore, deterministic scheduling achieves about 15% lower delay compared

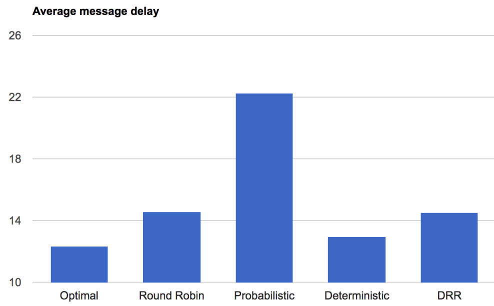

Figure 2: Intra-cell delay under near-symmetric traffic.

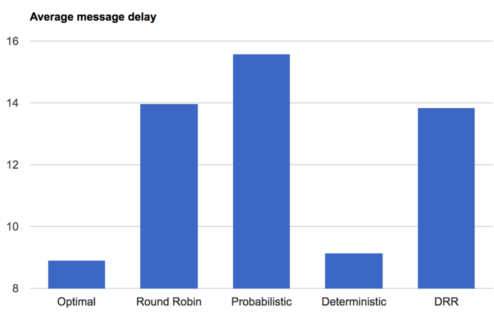

Figure 3: Intra-cell delay under highly asymmetric traffic.

to DRR. Also, we observe that the delay in deterministic scheduling is closest to the optimal delay, compared to other algorithms. The delay values (*i.e.*, on Y-axis) makes more sense if we compare them to 2 unit which is the time a pigeon (with unit speed) takes to travel from one end of the terrain to the other end. The result summary of the second experiment (*i.e.*, involving highly asymmetric cells) is reported in Figure 3, which compares the average message delay of the scheduling algorithms with the theoretical optimal delay. We again observe that deterministic scheduling gives better performance compared to the probabilistic approach and round-robin scheduling. Furthermore, deterministic scheduling achieves about 50% lower delay compared to DRR. Most interestingly, we observe that in this asymmetric condition, deterministic scheduling algorithm almost achieves the theoretical optimal delay (within 5% margin), which substantiates our theoretical framework.

4 RELATED WORK

In the current paper, we consider that all the foreign hosts and the home host in a cell share the communication channel (*i.e.*, the pigeon). However, not many prior works studied such a *shared channel* with the objective function of minimizing the *delay of messages* whereas they studied the *peer-to-peer mode*, which considered one-to-one communications among nodes, *e.g.*, [12] and the vehicle routing problems with pickup and delivery (VRPPD) [9],[8], [10]. Below we classify the body of related work.

Exact optimization. Psaraftis [8] considers the Dial-a-Ride problem, which is a VRPPD problem. Psaraftis proposed an *exact optimization* method inspired by the *dynamic programming* algorithm for the traveling salesman problem (TSP) in [2, 5]. However, [8] and the current paper differ in the mode of the resource demand: [8] considers discrete demand whereas we consider continuous demands. Furthermore, [8] considers the peer-to-peer mode rather than the shared channel mode.

Controlled mobility. Zhao *et al.* [12, 13] presented shortest TSP based heuristic algorithms for DTNs functioning in the peer-to-peer mode. Mancy *et al.* presented the DRR algorithm [7] which does not readily apply to the LFN scheduling problem. We have adapted this algorithm to our scenario and we compare its performance with ours in Section 3.5. Recently, [1] proposed a data synchronization scheme using a mobile relay to exchange data among isolated servers.

Partitioning based optimization. The research community explored partitioning-based heuristics to solve the TSP and the VRP problem. For instance, Karp [6] proposes a rectangular region partitioning (RRP) for the TSP. Furthermore, our prior work [14] partitions the cell in multiple sub-regions, and presents partitioning-based optimization of pigeon movement. However, we do not use any partition-based heuristics in our current paper.

5 CONCLUSION

We presented the design of LFN, which is a special type of DTN. One of the research challenges is to find the pigeon's optimum visiting frequencies in an LFN to achieve good networking performance (*e.g.*, message delay). We addressed this challenge by designing multiple scheduling algorithms. We compared the performance of the proposed algorithms with the state-of-the-art algorithms through simulation. Both theoretical analysis and simulation results confirmed the efficacy of our solution.

REFERENCES

[1] Kazuya Anazawa, Toshiaki Miyazaki, Peng Li, and Xiaoyan Wang. 2017. Big Data Synchronization among Isolated Data Servers in Disaster. In *GLOBECOM*. IEEE.

[2] R. Bellman. 1962. Dynamic Programming Treatment of the Travelling Salesman Problem. 9, 1 (1962).

[3] K Fall. 2003. A delay-tolerant network architecture for challenged internets.. In *Proceedings of the Sigcomm*.

[4] H Guo and J Li. 2007. Performance Analysis of Homing Pigeon based Delay Tolerant Networks. In *Proceedings of IEEE MILCOM*.

[5] M. Held and R Karp. 1962. A dynamic programming approach to sequencing problems. 10, 1 (1962).

[6] R Karp. 1977. Probabilistic Analysis of Partitioning Algorithms for the Traveling-Salesman Problem in the Plane. 2, 3 (1977), 209–224.

[7] Ahmed Mansy, Mostafa Ammar, and Ellen Zegura. 2011. Deficit round-robin based message ferry routing. In *GLOBECOM*. IEEE.

[8] H Psaraftis. 1980. A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. 14 (1980), 130–154.

[9] M Savelsbergh and M Sol. 1995. The general pickup and delivery problem. 29 (1995), 17–29.

[10] M Swihart and J Papastavrou. 1999. A stochastic and dynamic model for the single-vehicle pick-up and delivery problem. 114, 3 (1999), 447–464.

[11] A Vahdat and D Becker. 2000. Epidemic routing for partially-connected ad-hoc networks. In *Technical Report CS-200006, Duke University*.

[12] W Zhao and M Ammar. 2003. Message Ferrying: Proactive Routing in Highly-Partitioned Wireless Ad Hoc Networks. In *Proceedings of the IEEE Workshop on Futrure Trends in Distributed Computing Systems*.

[13] W Zhao, M Ammar, and E Zegura. 2005. Controlling the Mobility of Multiple Data Transport Ferries in a Delay-Tolerant Network. In *IEEE INFOCOM*.

[14] J. Zhou, S. Roy, J. Li, R. Hu, and Y. Qian. 2013. Minimizing the Average Delay of Messages in Pigeon Networks. *IEEE Trans. on Comm.* 61, 8 (2013).