Message Ferry Route Design for Sparse Ad hoc Networks with Mobile Nodes *

Muhammad Mukarram Bin Tariq, Mostafa Ammar, Ellen Zegura
College of Computing, Georgia Institute of Technology. Atlanta, GA. USA.
{mtariq, ammar, ewz}@cc.gatech.edu

ABSTRACT

Message ferrying is a networking paradigm where a special node, called a message ferry, facilitates the connectivity in a mobile ad hoc network where the nodes are sparsely deployed. One of the key challenges under this paradigm is the design of ferry routes to achieve certain properties of end-to-end connectivity, such as, delay and message loss among the nodes in the ad hoc network. This is a difficult problem when the nodes in the network move arbitrarily. As we cannot be certain of the location of the nodes, we cannot design a route where the ferry can contact the nodes with certainty. Due to this difficulty, prior work has either considered ferry route design for ad hoc networks where the nodes are stationary, or where the nodes and the ferry move pro-actively in order to meet at certain locations. Such systems either require long-range radio or disrupt nodes' mobility patterns which can be dictated by non-communication tasks. We present a message ferry route design algorithm that we call the Optimized Way-points, or OPWP, that generates a ferry route which assures good performance without requiring any online collaboration between the nodes and the ferry. The OPWP ferry route comprises a set of way-points and waiting times at these way-points, that are chosen carefully based on the node mobility model. Each time that the ferry traverses this route, it contacts each mobile node with a certain minimum probability. The node-ferry contact probability in turn determines the frequency of node-ferry contacts and the properties of end-to-end delay. We show that OPWP consistently outperforms other naive ferry routing approaches.

Categories and Subject Descriptors: C.2.1 [Network and Architecture Design]: Wireless Communication.

General Terms: Algorithms, Design, Performance.

Keywords: Delay & Disruption Tolerant Networks, Mobile Ad hoc Networks, Sparse Networks, Mobility Models.

*This works was supported by NSF awards CNS-0519784 and CNS-0313062 and a fellowship grant from NTT DoCoMo USA Labs.

MobiHoc'06, May 22–25, 2006, Florence, Italy.

1. INTRODUCTION

Mobile Ad hoc Networks (MANETs) are networks in which wireless mobile nodes cooperate to establish network connectivity and perform routing functions in the absence of infrastructure using self-organization [19, 12]. Since these networks do not require existing infrastructure and a priori planning, they can be rapidly deployed and have applications in a number of critical areas, such as, disaster relief, battlefields, and wide-area sensor networks.

Sparse Mobile Ad hoc Networks are a class of Ad hoc networks where the node deployment is sparse, and the contacts between the nodes in the network do not occur very frequently. As a result, the network can remain partitioned for extended periods of time. Several schemes for routing in Sparse MANETs exist, for example, [1, 3, 5, 10, 11, 14, 16, 26, 21, 23, 24, 29]. A common theme across all of these schemes is that they use a *Store-Carry-and-Forward* model, where the existing nodes in the network relay the data from the source to the destination nodes, in one or more hops, such that each node along the path receives the data from previous node and stores it locally. This node then carries the data for a while, and upon contact with other nodes, forwards the data. Schemes that rely on the intrinsic movement of nodes in the network for routing are referred to as "passive" schemes, yet others, where the nodes move proactively to bridge network partitions, are referred to as "active" schemes.

Data MULEs [21] and Message Ferrying [29, 30, 31] schemes are distinctively different from other active schemes for routing in the sparse networks, in the sense that in these models special mobile nodes called *Message Ferries*, or mobile agents called *Data MULEs*, are used for facilitating connectivity between nodes. The role of the message ferries[1] (or simply ferries), is to visit the nodes in the network and deliver the data among them. This is an attractive model because it allows serving a variety of networks and also allows achieving explicit end-to-end message delivery characteristics. Additionally, it takes the burden of message routing away from the nodes, which may be desirable when nodes have limited energy and storage resources.

A key problem under the Message Ferrying model is to design routes for the message ferries such that certain properties of end-to-end message delivery, such as, delay and loss, can be assured. This is a difficult problem when the nodes in the network move arbitarily. The difficulty in this context

[1]Data MULEs and Message Ferries are somewhat synonymous; we use the term message ferry for the work that we present here.

arises from the fact that as the nodes move arbitarily, we cannot be certain about their precise location at any given time, and consequently, we cannot easily design routes where the ferry can contact the nodes with certainty. In [21], the route of the MULEs is a random walk on a two dimensional grid, and the sensor nodes are considered to be stationary. In [29, 31], Zhao et.al., present a method for designing ferry routes when the nodes in the network are stationary. In [30], the same authors present a method for ferry route design when the nodes are mobile. However, they require that the nodes or the ferry move *proactively* to meet each other and in some cases require the use of long-distance radio to agree on a meeting location. Such communication may not always be feasible or desirable. Besides, such collaboration may disrupt the node mobility, which is undesirable as it may be dictated by non-communication needs.

Our key contribution through this work is an algorithm for designing message ferry routes in sparse networks where nodes have arbitrary movement. Our method *does not require any online collaboration between the ferry and the nodes, and does not disrupt the mobility of the nodes*. We call this method the Optimized Way-points (OPWP) ferry routing method. We determine message ferry routes that comprise an ordered set of way-points and waiting times at these way-points that are chosen carefully based on the mobility model of the mobile nodes in the network. Every time that the ferry traverses this route, it contacts every node in the network with a certain minimum probability. This node-ferry contact probability in turn determines the distribution of the node-ferry contact times, and ultimately the properties of end-to-end message delivery. The ferry can repeatedly follow the same OPWP route, without any changes, as long as the nodes continue to use the same mobility model, and the traffic demand does not change. Thus we do not require any kind of adaptability from the ferry once we program it to follow the route.

The rest of this paper is organized as following. In section 2 we review the related work. We describe the message ferry service model for mobile nodes in section 3, and the route design algorithm in section 4. In section 5 we present a comparison of the OPWP routing scheme with other routing schemes, and also present the system performance under different system parameters using simulation results. We conclude in section 6.

2. RELATED WORK

Sparse Mobile Ad hoc Networks are characterized by sparse node deployment and network partitions that may last for extended periods of time. Classical ad hoc routing protocols, such as, [19, 12, 20] are primarily designed for dense node deployments, where any partitions that occur are transient. These protocols do not suffice for sparse ad hoc networks.

Networking for sparse mobile ad hoc networks is also referred to as Delay or Disruption Tolerant Networking (DTN) [6]. A number of routing schemes exist for DTNs [1, 3, 5, 10, 11, 14, 16, 26, 21, 23, 24, 29]. These schemes exploit the fact that while an end-to-end path may not exist in the network at a given time, such paths do exist over time[2]. These schemes invariably use the *store-carry-and-forward* model for networking. Early work based on this model was Epidemic Routing [26] where the nodes exchange messages whenever they meet essentially flooding the network. Jain et.al. consider routing when the dynamics of end-to-end connectivity are known with certainty [10] or probabilistically [11]. All of these schemes use the intrinsic mobility of the nodes in the network.

[2]Sparse networks have also been referred to as "Space-Time" connected[18, 10], or Partially connected[26] networks.

Another set of work considers the possibility of controlled mobility for DTN routing. Li and Rus[16] consider algorithms for optimally and proactively moving the existing nodes in the network for routing between disconnected parts of the network, but they require online collaboration between the nodes. Shah et.al. in Data MULEs project [21] propose using wireless mobile agents that move randomly on a grid to facilitate connectivity with sparsely deployed stationary sensors in the network. In a similar work, Zhao et.al.[29, 30, 31] and Somasundara et.al.[22] proposed the communication models where a special mobile node facilitates network connectivity between the nodes in the network. Kansal et.al.[13] make a case for controlled mobility for acheiving desired properties in the network, such as, controlling delay, and energy consumption. Luo and Hubaux[17] propose that the sink node in a mesh network setting to be mobile, so that the load of message forwarding toward the sink node is more uniform across the nodes in the network; this increases the lifetime of the network. Gu e.al.[7] consider the ferry route design problem when the nodes have different service requirements.

Our work can be considered complimentary to the earlier work, as we extend the message ferrying model so that the ferry can contact the mobile nodes *without online collaboration*. Recently, Leguay et.al. [14] proposed building "mobyspace", or a map of probabilistic locations of the mobile nodes, and then using this information to choose the next hop which is closest to the destination. Our work is distinct in that we use the probabilistic node location to construct routes for the message ferry.

3. MESSAGE FERRYING MODEL

In the Message Ferrying model, the devices in the network are classified into two categories [30]. (i) Regular nodes, or simply the nodes, that move according to some mobility model. These nodes generate data for other nodes in the network in the form of application layer data units called *messages*. At the same time, these nodes are interested in receiving the messages that other nodes have generated for them. For this work we assume that all the messages are unicast, i.e., they have a single unique destination. We assume that the movement of the nodes is driven by non-communication needs (e.g., a field-task assignment), and therefore this movement cannot be disrupted. (ii) A single special node called message ferry (MF) that is responsible for delivering the messages between the nodes. The ferry achieves this by traversing a predetermined path *repeatedly*. We refer to each traversal through this route as a *tour*.

We assume that both the ferry and the nodes are equipped with a similar radio of given small communication range. The nodes and ferry can communicate with each other only when they are within a distance of each other that is less than the communication range. The node and ferry are said to be in *contact* when they are within the communication range of each other. We assume limited communication range because nodes may be energy constrained and may not be able to use long range communication channels that may require more power. Furthermore, while the ferry may be able to use a long range radio, the range of two-way communication between the node and the ferry would still be limited by the communication range of the nodes. Our model requires two-way communication for contact establishment. Similarly reliable data transfer between the nodes and the ferry, such as, using TCP, may also require two-way communication.

During each successful contact, the ferry exchanges messages with the nodes. The ferry *uploads* the messages that the node has generated for other nodes, and *downloads* the messages that the ferry has for the particular node. The process is referred to as *service*. The ferry services only one node at a time.

In the time between successive contacts, the nodes store the messages that they generate in a local buffer, called *send buffer*. We assume that the send buffer can hold a certain maximum number of messages, and once the send buffer is full, any new messages that the node generates are lost. We also assume that each node has a similar buffer for the messages that it receives; we call it the *receive buffer*. The receive buffer is used to store the messages that the node receives until they are consumed by the application layer at the node. We assume that the receive buffer for a node can hold a certain maximum number of messages.

As the ferry takes the tour, it meets with different nodes. Upon meeting with a node, the ferry begins the download service, and continues until the ferry has downloaded all the messages that it has for the node, or, a timer, which we call the *download timer*, expires, or the receive buffer of the node becomes full, whichever occurs first. The ferry attempts to deliver any messages for the node that are left in its buffer at the end of download service in the next contact with the node.

After the download service, the ferry starts the upload service. The ferry uploads the messages from the source buffer until all the messages in the source buffer of the node are uploaded, or a timer, called the *upload timer*, expires, whichever occurs first. Any messages that are left in the send buffer remain buffered until the next contact with the ferry. We refer to the messages that are left in the send buffer of the node as the *residue* messages. Please note that our model does not force strict order that the download service precede the upload service. These could happen simultaneously (if the radio channel permits), or in some multiplexed fashion. In general, performing download before upload reduces average delay, albeit slightly.

Since both the ferry and the nodes are mobile in our model, we make a simplifying assumption that when the ferry and a node come in contact with eachother, either the contact lasts long enough to complete the service, or they can pause, and exchange messages; usually the service time is short and does not amount to disruption in node mobility. After the exchange the ferry continues with its route and the node continues with its movement. We assume that the ferry has infinite resources to move around and meet the other nodes, as well as to communicate with the other nodes when they come within its radio range, and to carry the messages between the nodes. Furthermore, we assume that we can route the message ferry in whatever way we want in the region where the nodes move. In this work the only constraints that we consider regarding the ferry are that the ferry cannot move faster than a certain maximum

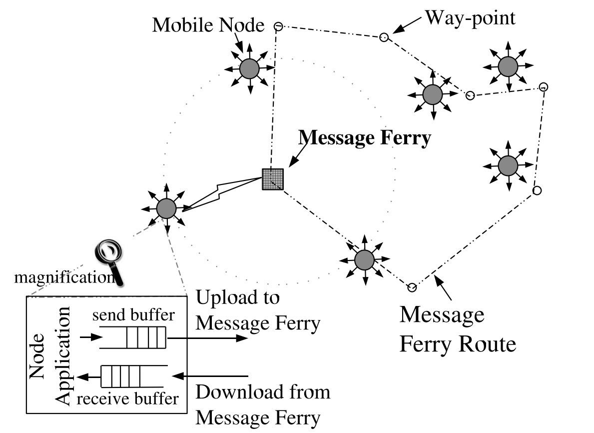


Figure 1: An Illustration of Message Ferrying Model

speed, and that it can only communicate with other nodes that are within a given radio range of the ferry.

4. ROUTE DESIGN

Our goal is to design routes that the ferry can follow repeatedly, and that do not require any online collaboration between the ferry and the nodes. At the same time we require that the nodes and the ferry meet frequently so that desired message delivery characteristics are achieved. To achieve this, we produce ferry routes that are in the form of an ordered set of *way-points* and *waiting times* at each of these way-points. These are chosen carefully, so that each time that the ferry takes the *tour*, it meets each node with a certain minimum predetermined probability. This probability in turn governs the frequency of node-ferry meetings, and that ultimately determines the characteristics of end-to-end message delivery. In this section we will describe the process of designing the message ferry route. We start with describing the key idea behind this process. Later we describe the algorithm in detail.

4.1 Key Idea

The main difficulty in designing ferry routes for arbitrarily moving nodes is that we cannot correctly predict the location of the nodes, and hence it may not be possible to correctly and deterministically position the ferry to contact the nodes. Observe that if there exists a steady-state in the mobility model, such that the steady-state probability of presence of a mobile node in a specific sub-region (such as one defined by radio coverage area around a ferry), is non-zero, then we can contact the node with certainty if we wait for the node in that sub-region for long enough. However, for most mobility models, this probability approaches 1, only as the waiting time approaches infinity. Clearly, we cannot afford to wait for infinitely long, and hence, we cannot afford to contact the nodes with certainty.

However, it is possible to meet the nodes with a desired probability by waiting a finite amount of time at a sub-region, as long as the desired meeting probability is modest (i.e., large but not approaching 1), and the steady state probability of node presence in that sub-region is substantial (i.e., not approaching zero). If the ferry is able to contact a particular node with some probability $0 < p \leq 1$, once in every tour, then in expectation the ferry will meet the node in every $1/p$ tours. More specifically, the number of tours that the ferry may make between successive node-ferry contacts is a geometric random variable with probability p. This observation is key to the ferry route design algorithm that we present in this paper.

The ferry route design process consists of two steps. In the first step we choose the *way-points* and the waiting times at these way-points so that the ferry can meet each node with certain probability. In the second step we order these way-points together to form a tour that is of minimum length. The time that it would take the ferry to go take the tour is the sum of time that it takes to travel between each of the way-points, the waiting time at each of the way-points, and the time that the ferry spends to service the nodes. For simplicity, let's assume that it takes some constant T units of time for the ferry to complete the tour, and in each tour, it meets each node exactly once with probability p, then the expected time between successive contacts is T/p. [3]

Message delay and message loss depend on the frequency of node-ferry meetings, or the time between successive node-ferry contacts. Thus we aim to reduce the expected time between contacts, or the T/p ratio. We can reduce T, or increase p, or do both. However, it turns out that reducing the ratio T/p is not trivial. In order to minimize T, we not only need to choose the way-points that would require minimum waiting time, but would also yield shorter overall tour. So we have to optimize the two steps in the ferry route design together. Moreover, reducing T/p presents a trade-off. Since the contact probability p increases only as the waiting time increases, increasing p may inadvertently result in increasing the overall tour duration T.

In the following we describe the ferry route design algorithm in more detail. We start by the modelling the node mobility and presence and later describe the route design algorthim that uses these models.

4.2 Mobility and Node Presence Modelling

We consider a finite, two dimensional space S in which the nodes and the ferry move. We define C, the set of *candidate way-points*, as a small subset of points in S. Specifically, we draw a grid of some resolution on S and pick the centers of the grid squares as candidate way-points. The ferry route way-points set, R, comprises an ordered subset of C. $W = \{w_s : s \in R\}$ is the set of corresponding waiting times on the chosen way-points. We assume that the ferry moves from one way-point to another with speed v_f in a straight line. We assume that the nodes move independent of the ferry. Consider r to be the communication range of the nodes and the ferry, then the tuple $\Omega = (R, W, v_f, r)$ completely specifies the ferry route.

Consider $F(t)$ as the location of the ferry at time t, and $N_i(t)$ as the location of the node i at time t. Also consider $a(s, r)$ as a circular region of radius r with center at s. We say that $N_i(t) \in a(s, r)$, if node i is within distance r of point s. Let's suppose that the ferry sends a periodic beacon every Δ seconds to announce its presence. If a node is within the communication range r, then it will receive the beacon, and reply immediately, allowing a contact to occur.

Suppose that the ferry tour starts at a time t_0, and the ferry sends a beacon every Δ seconds from t_0, then a contact would occur if for some $k \leq T/\Delta$, $N_i(t_0 + k\Delta) \in a(F(t_0 + k\Delta), r)$ (here T is the total duration of route Ω). When both the node and the ferry are moving, analyzing the probability of existence of such k becomes very difficult. We divide the beacon times into two sets; K_T the set of beacons sent by the ferry when it is moving, and K_W the set of beacons sent when the ferry is waiting at a way-point.

Consider, $G_i(\Omega, t_0) = \text{Prob}[\exists\, k \in K_T : N_i(t_0 + k\Delta) \in a(F(t_0 + k\Delta), r)]$, i.e., the probability that a contact with node i will occur when the ferry is moving; we call it the *instantaneous contact probability*. Also consider, $H_i(\Omega, t_0) = \text{Prob}[\exists\, k \in K_W : N_i(t_0 + k\Delta) \in a(F(t_0 + k\Delta), r)]$, or the probability that the node will arrive in the communication range of the ferry when the ferry is waiting at a way-point; we call it the total *time-cumulative contact probability*. If we want a route where the probability of contact with node i is p_i, then we want a route specification Ω, such that $G_i(\Omega, t_0) + H_i(\Omega, t_0) > p_i$ for all nodes i. This dissection is still not very tractable for general node mobility. We make some assumptions that make our framework more useful.

First we assume that the time-limiting instantaneous and the time-cumulative contact probabilities exist for the entire route Ω, as $G_i(\Omega)$ and $H_i(\Omega)$[4], or we are able to at least provide a lower bound for these probabilities. We also assume that such probabilities exist for individual candidate way-points in space. We denote the time-limiting probability of instantaneous contact with node i if ferry is located at a point s as, $g_i(a(s, r))$. Similarly, we denote the time-limiting time-cumulative contact probability with node i, if the ferry waits at point s for w_s units of time, as $h_i(a(s, r), w_s)$. Since both the nodes and the ferry are mobile, they may keep missing each other as ferry moves from one location to another. If the ferry visits all the locations in set R then the probability that the ferry instantaneously finds the node i in at least one of the locations $=$ $\Pr\{\bigwedge_{s \in R}$ Ferry meets node i instantaneously at location $s\} \geq$ $\max_{s \in R} \Pr\{$Ferry meets node i instantaneously at location $s\}$. Since $g_i(a(s, r))$ is the probability that the ferry instantaneously finds the node i at location s, therefore, the probability of instantaneous meeting with node i in the route Ω is $G_i(\Omega) \geq \max_{s \in R} g_i(a(s, r))$. Note that there is also some probability of instantaneous meeting while the ferry is moving from one location to another, but we ignore that for simiplicity. Overall, we are underestimating the probability of instantaneous meetings, and we might incur some penalty for this underestimation. However, as we will show through simulation results later, this effort is worth-while despite the simplified model.

We, assume that h_i are independent, and we define $H_i(\Omega) = \sum_{s \in R} h_i(a(s, r), w_s))$. In order to meet node i with probability p_i, we require a route $\Omega(R, W, v_f, r)$ such that:

$$\begin{aligned} G_i(\Omega) + H_i(\Omega) &\geq p_i \\ \max_{s \in R} g_i(a(s, r)) + \sum_{s \in R} h_i(a(s, r), w_s) &\geq p_i \qquad (1) \end{aligned}$$

The framework that we have just described will work for any mobility model for which we can determine h and g. In

[3] This is a simplifying assumption that we make for the purposes of modelling. If multiple node-ferry contacts may occur in a tour, then T/p is the expected upper bound for the time between successive contacts. For the results that we present later in the paper, we do not assume that the time for the tour is constant, and we also allow multiple contacts between a node and the ferry during each tour.

[4] We drop subscript for time from our notation for the time limiting probabilities.

the following, we describe how these functions would look like for three different mobility models.

1. Stationary or Limited Mobility: If the movement of node i is limited, such as, when it is non-mobile, or moves in such a small area that it can always be found in $a(s, r)$ for some s and given r, then $g_i(a(s, r))$ is 1. We define $h(a_{s,r}, w) = 1$ for any $w \geq 0$ for such s.

2. Periodic Movement: If a node i moves along a fixed path with a period α, then we can meet that node with certainty by waiting for time α at a point that is along its path. If we wait for w time, beginning at a random instance, at a point s, that is along the path of the node, then

$$h_i(a(s, r), w) = \begin{cases} w/\alpha & 0 \leq w \leq \alpha \\ 1 & w > \alpha \end{cases} \quad (2)$$

That is, the time cumulative meeting probability increases linearly with waiting time, until it reaches 1, after which it remains 1. The time cumulative probability is zero for any w, if $a(s, r)$ does not intersect the path of the mobile node. If the node travels at a constant speed, then the max $g_i(a(s, r))$ is $g_i(a(s, r))$ for a point s that gives maximum intercept of node path for $a(s, r)$.

3. Random Way-point Mobility Model: Using simulation traces we have found that if a node i follows the Random Way-point model in a bounded region S_i, such as, a square, then the probability of node's arrival in a region in the next t seconds, beginning at a random instance is exponentially distributed. In particular,

$$h_i(a(r, s), w) \approx 1 - exp(-w/\gamma_{s,r}) \quad (3)$$

The mean inter-arrival time, $\gamma_{s,r}$, depends on the size (r) and the location (s) of the region a. Larger r typically yields smaller $\gamma_{s,r}$ for given s. We also found that for given r, $\gamma_{s,r}$ would be smaller if s is a point closer to the center of S_i, and significantly larger if s is a point that is farther away from the center of S_i. The maximum $g(a(s, r))$, occurs at a point in the center of S_i, however, its value is very small. We have observed that the distribution is stable, and the arrivals are memoryless to a oblivious ferry (one which assumes no knowledge about next random way-point of the mobile node, or its speed). At this point, we do not have a closed form expression for estimating $\gamma_{s,r}$, or $g(a(s, r))$ for the RWP model. Instead, we use empirically drived values for the simulations that we present in this paper. We provide detailed information about how we estimate these parameter for RWP model in [25].

We emphasize that our framework is applicable to any mobility model where we can determine g and h. As we described above, we were able to determine g and h for the Random Waypoint model using simulation generated mobility traces. We believe that these distributions could be found for other mobility models as well using similar techniques. Jain et.al. [9] have recently shown using empirical data from a university campus environment, that the user registration patterns follow a Weibull distribution. While they do not present a mobility model, their observation shows that if there were a mobility model that was representative of real-life movement, we will be able to model its desired characteristics reasonably well.

For the present work, we assume that $h_i(a(s, r), w)$ is always convex (or concave). This is a reasonable assumption for any mobility model where the probability of arrival does not decrease with increasing waiting time.

Grid size for the candidate way-points is an important issue. It essentially serves to "discretize" the node presence and mobility process, which is otherwise a continuous process. With higher granularity grids, we have more accurate information about node presence and mobility, and hence we can choose way-points and waiting times more accurately. However, as the candidate waypoints increase, so does complexity of choosing a subset for the eventual ferry route.

4.3 Ferry Route Design Algorithm

The two steps in ferry route design are: (i) finding a good set of way-points and their corresponding waiting times, and (ii) ordering these way-points together to form a tour. We first look at the two steps independently.

4.3.1 *Choosing Appropriate Way-points*

Suppose that we somehow decide that we want the ferry to contact the node i with probability p_i in each tour (we address the question of how to decide on p_i a little later in the text). If we have the knowledge of the instantaneous contact probability function $g_i(a(s, r))$, and time cumulative probability functions $h_i(a(s, r), w)$ for every node i, and for every point s in the candidate way-points set C, then our next step is to choose waiting times w_s corresponding to each way-point s, so that the total meeting probability for node i is at least p_i. Clearly, we want to minimize the total waiting time. Our optimization problem is following.

Optimization Problem 1

$$\min_{\Omega(R,W,v_f,r)} \sum_{s \in R} (w_s)$$

subject to:

1. $\forall$ nodes i

$$\max_{s \in R} g_i(a(s, r)) + \sum_{s \in R} h_i(a(s, r), w_s) \geq p_i$$

2. $\forall \; s \in R$

$$w_s \geq 0$$

An interesting aspect of the above problem is that when the regions of movement of some nodes overlap, and a way-point in the overlapping region is chosen, then we can use the same waiting time to wait for multiple nodes simultaneously, thus reducing the overall waiting time. We call these *shared way-points*. Two or more nodes may show up simultaneously at such a way-point and in such an event the ferry chooses to serve one of them. For the results that we present in this paper, the ferry chooses the least recently served node. If the ferry is already serving a node, when another node arrives, then the ferry continues to serve that node. In general, such events are very rare in sparse deployments.

If the node for which the ferry is waiting at a way-point arrives before expiration of the waiting time, then if the way-point is not shared, the ferry stops waiting and starts moving to the next way-point immediately after servicing the node. If however, the way-point is shared, then the ferry only stops waiting if it has already contacted all the nodes in the present tour that share the way-point, or the waiting time expires. As an optimization, the ferry completely skips a way-point if it has already met all the nodes in the present tour for which it was supposed to wait for at that way-point.

4.3.2 Constructing a Path from Way-points

Once we have determined the way-points, we order them so as to minimize the length of the route. This amounts to the planar travelling salesman problem (TSP) whose exact solution is NP-hard. TSP solvers like Concorde [4] can solve the problem exactly for few hundred points within minutes. If the number of points is large, then we can opt for any of the available approximation algorithms that exist for planar TSP. We refer the reader to [27] for further details.

Note that while the solution to the problem for finding the way-points will result in minimizing the aggregate waiting time at the way-points, the solution may not be suitable for minimizing the length of the route. As an example, consider a solution in which there is a way-point that is far-away from the rest of the points, and is chosen because a small amount of waiting at that way-point contributes significantly to contact probability. However, the additional time required by the ferry to travel to such a way-point may negate the advantage. Instead, there might be a point that is closer to the rest of the way-points at which the ferry might have to wait longer to gain a similar contribution to contact probability, however, since this point is closer, we save travelling time for the ferry, and can thus compensate for the longer wait.

Ideally, we would like to solve the problem in a way that reduces the sum of the waiting time and the travelling time. However, the complexity of such approach would be immense. Most likely we cannot find the travelling time without first enumerating the desired way-points. Thus to find the optimal solution to such a problem would require enumerating all feasible solutions to way-point problem and then picking one with minimum total waiting and travelling time.

We introduce a heuristic that promises to work well under our assumptions. Our intuition is that if the way-points are close to eachother, then they yield a shorter route. If the regions of the mobile nodes are spread more or less uniformly across the universal region S, then we would prefer points that are closer to the center of the region S. Based on this intuition, we modify the optimization problem that we presented earlier, so that now we try to minimize a weighted sum of the waiting time and the total distance of selected way-points from the center of the region. For all $s \in C$, consider l_s as the absolute distance to the center of S, then we formulate the problem as following.

Optimization Problem 2

$\min_{\Omega(R,W,v_f,r)} \sum_{s\in R}(w_s + \beta\, l_s)$

subject to:

1. $\forall$ nodes i

   $\max_{s\in R} g_i(a(s,r)) + \sum_{s\in R} h_i(a(s,r), w_s) \geq p_i$

2. $\forall\ s \in R$

   $w_s \geq 0$

Here β is a constant, whose value determines the relative importance of goal of finding shorter paths and smaller overall waiting time in the tour. Suitable values for β tend to depend on the units of distance as well as the average speed of the ferry. We use a value of 0.1 in our results.

Once we have chosen the appropriate way-points, and ordered them to form a path, we obtain a message ferry route. Note that the total time T, that the ferry takes to complete the route has three components (i) Waiting Time in the Route (T_w): The sum of waiting times at the way-points, (ii) Journey Time in the Route (T_j): The total time that the ferry spends travelling between way-points, and (iii) Service Time in the Route (T_s): The total time spent in servicing the nodes that the ferry meets along the way. Total route time $T = T_w + T_j + T_s$. So far in this section we have talked about the waiting times and journey time, now lets discuss some properties of the service time.

The service time depends on the service discipline. As described earlier in section 3, the service to each node that the ferry contacts in a tour has two components: (i) The upload service, and (ii) the download service. The service time in a particular tour would depend on the message arrival rate and the time since last service, but it is always time-limited by the values of download timer D_i and upload timer U_i. The expected worse case service time in route is $\sum_i^n p_i(U_i + D_i)$, where n is the number of nodes. While U_i and D_i are parameters of the system, and can be set to arbitrary values, a sensible choice is to set these timers such that the total service time is never less than the total service time required to service messages that are generated by the node between successive contacts. For example, if the node i generates λ_i messages per second, then in expectation, it would generate $\lambda_i T/p_i$ messages between successive contacts, so the upload timer should be $U_i > (\lambda_i T)/(p_i B)$, where B is the bandwidth of the channel between the nodes and the ferry, measured as service rate or number of messages uploaded or downloaded per unit time. Similarly, if the aggregate traffic destined to a node i is κ_i messages per unit time, then the download timer should be set to $D_i > (\kappa_i T)/(p_i B)$. Without these settings, the queues at the sender or ferry could grow indefinitely (send buffer is limited, so while the queue would not grow indefinitely, there will be persistent loss).

4.4 The T/p Trade-off

In section 4.1 we discussed that reducing the expected time between successive contacts (T/p) presents a trade-off. Here we examine this trade-off in detail.

Since we are considering sparse deployments, typically, the duration of the tour T contains a large journey cost (in terms of time) T_j, which depends on the ferry speed, and if the ferry speed is relatively small, then T_j constitutes a large portion of overall tour time. As a result, we want the ferry to contact the nodes with as high a probability as possible in each tour, as that would amortize this cost. However, the other two components in route time, T_s and T_w depend on p. The service time T_s seems to decrease linearly with increasing probability. However, if the contact probability is primarily due to waiting, as we discussed in section 4.2, then the T_w will increase as we try to increase p.

The exact nature of trade-off depends on the time-cumulative probability function for a specific node. We discuss the cases where the time cumulative probability function is exponential or linear.

4.4.1 Exponential Time Cumulative Function

Recall that the CDF of a exponentially distributed random variable is concave, so we would expect that for some range of waiting time the contact probability increases more rapidly than the waiting time (thus a possibility that T/p would infact decrease as we increase p) and at the same time there is a range of waiting time values, for which p increases slower than the waiting time, implying that T/p

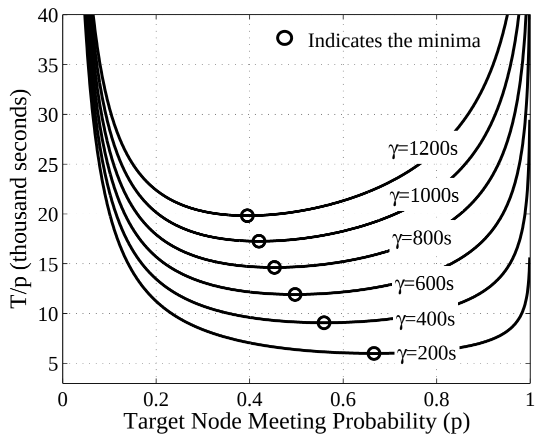


Figure 2: The T/p trade-off for the exponential time cumulative probability function. (T_j = 1000 sec, n=10, γ is the mean of exponential distribution)

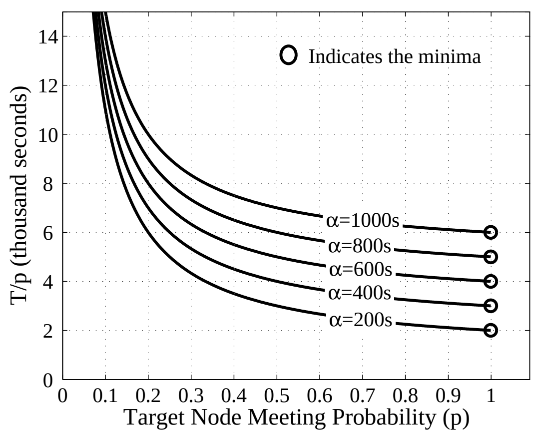


Figure 3: The T/p trade-off for linear time cumulative function. (T_j = 1000 sec, n=10, α is the slope of time-cumulative probability funciton)

would increase for those values of p. This suggests that T/p is a convex function of p, and there is an optimal p, that minimizes the ratio.

To better understand this trade-off, let's make some simplifying, but not necessarily non-realistic assumptions. Assume that $T_s \ll T$, so that its affect is minimal; this would be true if the data-rate between ferry and node was high compared to the message arrival rate at the source nodes. Suppose that we have n mobile nodes, which all use identical mobility models and their regions do not overlap. Suppose that instantaneous contact probability ≈ 0, this would be true for most large regions. Suppose that we want to meet all the nodes with the same probability p, and that we end up with a route that has a single way-point corresponding to region of each mobile node, and the waiting time at each of these way-points is the same. In this case T_w is n times the waiting time that is necessary to achieve p; let's call this required waiting time t_w. Under these simplifying assumptions, $T = T_j + nt_w$, and $p = 1 - exp(-t_w/\gamma)$; here γ is the mean inter-arrival time for the region defined by the communication area of ferry at each of the chosen way-points.

$$\left(\frac{T}{p}\right) = \frac{T_j + nt_w}{p} = \frac{T_j - \gamma n \ln(1-p)}{p} \quad (4)$$

We can easily obtain the exact minima for equation 4, however, it is more instructive to visually inspect the shape of this function. Figure 2 shows that while the function is indeed convex, it is fairly flat at the bottom. This is quite remarkable, as it indicates that even if we do not choose a precisely optimal target meeting probability p, we will continue to do reasonably well as long as we do not choose a p that is either too small or too large. The function behaves in a similar manner for different number of nodes (n); larger values of n result in higher T/p, but the shape of the curve remains convex as in figure 2. For given γ and n, the optimal p increases as we increases T_j, as expected.

4.4.2 Linear Time Cumulative Probability Function

Figure 3 shows the trade-off for the case when the time-cumulative probability increases linearly with waiting time, as in the case of periodic movement that we described earlier. Note that optimal p is 1, regardless of α, because here T_w increases linearly with p, and any increase in p, only amortizes the cost of the tour.

4.5 Possible Enhancements

4.5.1 Service Differentiation

We can provide service differentiation among the nodes by configuring the upload and download service timers appropriately for each node. A larger upload timer results in fewer residue messages and lower loss probability. Larger upload timers can also be used in cases where a node generates proportionately more traffic than other nodes. Larger download timers are also useful for nodes which attract more traffic than others; with higher download timers more traffic can be downloaded in fewer successful contacts, thus reducing delay.

Contact probability can also be used for service differentiation as higher contact probability in each tour means that the ferry meets the node more frequently, reducing the delay and the message loss. However, contact probability cannot be increased indefinitely, because as we mentioned in section 4.4, T/p presents a tradeoff. For the nodes requiring higher service level, we choose the optimal contact probability, and for the nodes that require lesser service level, we may choose smaller contact probability by choosing a smaller wait time.

We note that Gu et.al.[7] propose a scheduling scheme where the mobile agent visits certain nodes more frequently depending on the amount of data that they generate, providing a form of service differentiation.

4.5.2 Adaptability for Joining and Leaving of Nodes

Our work mostly assumes that there is a fixed number of nodes in the ad hoc network. Here we sketch a possible method for making the framework adaptive to nodes leaving and joining the network.

We set a threshold, Z_i, such that if the ferry does not manage to establish a contact with node i in Z_i consecutive tours, then the ferry assumes that either the node has gone down, or its mobility model has changed. In such a case the ferry removes the node from its list and recomputes its route. If the ferry meets the node in the future, or learns through external means that the node exists, then the ferry acquires the node's mobility model and re-incorporates it into its route. Similarly, if the ferry meets a new node, that it does not know, then it acquires the nodes presence and mobility model and incorporates it into the route. The ferry could also adapt to changed traffic patterns by appropriately setting the upload and download service timers for the individual nodes based on the traffic patterns.

5. SIMULATION RESULTS

5.1 Simulation Setup

We choose a setup with small number of nodes based on the premise that the node deployment is sparse. In particular, we consider a setup with 10 mobile nodes ($n = 10$) that move in a 4km x 4km area according to one of the following mobility models. Please note that our framework can easily accommodate more number of nodes, and the expected time between successive node-ferry meetings increases slower than linearly with the number of nodes as shown in eq. 4.[5]

1. Random Way-point (RWP): Each node moves according to the random way-point mobility model in the entire 4km x 4km region. The RWP parameters are same for all nodes; $v_{min} = 9m/s$, $v_{max} = 11m/s$, and an exponentially distributed pause time with mean of 1 sec. We choose a large v_{min} as otherwise RWP model fails to provide a steady-state in terms of average speed [28].

2. Area Based Random Way-point (AB-RWP): Each node moves according to the random way-point mobility model in a 1km x 1km region whose center is uniformly distributed in the 4km x 4km area. These regions are shown as dotted rectangular boxes in figure 4. The RWP parameters are the same for all nodes; $v_{min} = 9m/s$, $v_{max} = 11m/s$, with exponential pause time with mean of 1 sec.

3. Circular Motion (CIRC): Each node moves in a circular path of radius 500 meters, in a 1km x 1km region whose center is uniformly distributed (such as shown in figure 4). The circle is centered at the center of the region. Initially, each node starts at a random point on its circular path. The node moves at a constant speed chosen randomly between $v_{min} = 9m/s$, $v_{max} = 11m/s$, at the start of each loop through on the circle. At the end of each round through the circle, it pauses for an exponentially distributed time with mean of 1 second, before choosing a new speed and continuing with its movement.

We assume that each node generates messages at a rate of λ messages per second, and each message is destined for one of the other $(n-1)$ nodes uniformly, at random. The nodes consume the messages that they receive from ferry at a rate of μ messages per second. In particular, we take $\lambda = \mu = 0.1$. We also assume that the nodes have a send buffer and a receiver buffer that can each hold 1000 messages. We assume that the radio communication range is 100m for contact discovery as well as for sending and receiving messages to and from the ferry. We also assume that the node and ferry can exchange messages at a rate of 100 messages per second when they are within the radio range and a contact has been established. Lastly, we assume that the node and the ferry learn about presence within the radio range using a beacon that is repeated every 2 seconds. An immediate effect of this periodic beaconing is that if the ferry and node temporarily come within the radio range, but there is no beacon during that time, then the ferry and node will fail to meet.

[5]The journey time, T_j, is sensitive to node deployment. In particular, we found that the tour length increased slower than linearly as we increased the density of nodes for uniform random node deployment. Total waiting time may also increase slower than linear, as higher density may result in overlapping of mobility regions and more shared way-points.

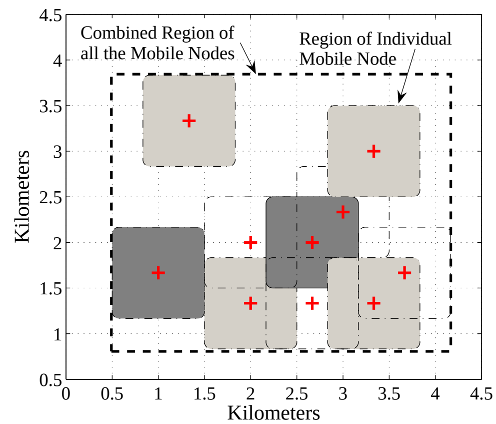


Figure 4: Simulation setup: regions of the mobile nodes. The + sign indicates the center of the region. The regions are shaded for better presentation.

We assume that there is a single ferry in the system that moves according to one of the following five way-point based ferry routing models. We assume that in each case, the ferry moves repeatedly from one way-point to the next. For each leg, it chooses a speed randomly between $v_{min} = 9m/s$, $v_{max} = 11m/s$. This allows us to have some variability in ferry's journey time. Note that increase in ferry speed results in a linear decrease in journey time, and may also result in a shift in optimal contact probability (ref: eq.4). In general, higher ferry speeds will result in smaller inter-contact times and better performance. Later in this section we discuss the impact of ferry speed on the system performace.

1. Random Way-point (RWP): The ferry moves in the larger, (4km x 4km) region **S** according to the random way-point mobility model. At each way-point, the ferry pauses for exponentially distributed time with mean of 1 sec.

2. Restricted Random Way-point (RWP-res): The ferry moves as in the ferry RWP model, with the exception that the way-points are only chosen from the part of the region which is also part of mobility region of one of the mobile nodes. We expect this model to work somewhat better than the vanilla RWP ferry route model. Note that this ferry model can also be thought of as one where the ferry visits the region of one node after the other, at random.

3. Space Filling Way-point (SFWP): In this model, we divide the 4km x 4km region into a grid, and pick the center of the each grid-box as a way-point. Next we order these way-points so as to form a shortest possible tour using the Concorde Travelling Salesman (TSP) solver [4]. The ferry route comprises traversing this ordered set of way-points repeatedly. We denote the model where the grid size is X times the communication range as SFWP-X model.

4. Region Centers Way-point (RCWP): We pick the centers of the regions of each of the mobile nodes as the way-points and order these using the Concorde TSP solver. The ferry route comprises traversing this ordered set of way-points repeatedly.

5. Optimized Way-points (OPWP): We pick way-points and waiting times at each way-point according to the algorithm that we described in 4.3. When the nodes follow the AB-RWP model, the optimal target probability turned out to be ≈ 0.6. For the CIRC node mobility model, the algorithm suggests target probability of 1 as CIRC has linear time-cumulative probability function. In the case where nodes follow the RWP model, the OPWP algorithm gives a route comprising a single point, as expected. We have used a custom solver for solving the optimization problem where it involved exponential functions. It solves the problem over 1600 candidate way-points (a grid-size of 100m), using a combination of interior point method for solving convex optimization problems [2] and some manual guidance to speed up the search.

When a contact occurs, and the ferry decides to service a node, the two stop and until the service is completed. We have set the upload and download timers to 6 seconds each. For this simulation, we allow the contacts to occur at any time, whether the ferry is waiting for a node at way-point or not at the time instance when the node-ferry contact occurs. Furthermore, we allow that a node may meet a ferry multiple times within a tour as long as the contact with the occurs at least 30 seconds after the last service for that node. This constraint ensures that we benefit with all the contact opportunities that are afforded, while the minimum time between successive contacts allows us to keep the service discipline to be time-limited. Without this constraint, the node and ferry may never move away from eachother, because soon after completion of a service episode the node and ferry may realize that they are still with-in the radio range of eachother and start a new service episode.

For the present work we consider that the nodes only communicate using the ferry as an intermediary so that we can study the impact of different ferry route design algorithms in isolation, without having to worry about any artifacts that may arise due to peculiarities of specific ad hoc routing methods. This arrangement does not impact our results in any significant way. If we allow the nodes to communicate directly with eachother, then a concern is that the communication between the nodes may nterfere with the node-ferry communication, and result in loss of data or missed contacts. However, in the sparse settings that our work aims to address, the probability of other nodes in the vicinity during a node-ferry contact is minimal. We report some results at the end of next section.

5.2 Performance Comparison

We compare the the different ferry routing models based on the following three criteria.

1. Frequency of Node-Ferry Contacts: Our ultimate goal in ferry route design is to contact the mobile nodes as often as possible. Under this criteria, we evaluate the ferry routing schemes based on how frequently the node-ferry contacts occur over all, across all the nodes.

2. Fairness of Node-Ferry Meetings Among Nodes: Fairness is an important concern in node-ferry route design. Ideally we want that the ferry contacts each node equally often (when the demands are uniform) or in a way that is proportional to the demand. Note that without this requirement, we could design a ferry route where the ferry only targets a small subset of nodes, thereby achieving a high frequency of node-ferry meetings, but the other nodes in the network starve. For our simulation setup, the demands are uniform, as the destination node is chosen uniformly among all the possible nodes, and message arrival rates are the same for all the nodes, thus we can use the unweighted Raj Jain fairness index [8] for comparing the different ferry routing schemes. If ω_i is the number contacts per unit time for node i, then the fairness index is defined as:

$$f(\omega_1, \ldots, \omega_n) = \frac{\left(\sum_{i=1}^{n} \omega_i\right)^2}{n \sum_{i=1}^{n} \omega_i^2}$$

A fairness index value close to 1 would indicate good fairness, whereas values significantly less than 1 will indicate lack of fairness.

3. Delay and Loss of Messages: Although we are in the delay tolerant networking paradigm, the delay and loss are still metrics of interest. We compare the end-to-end delay, defined as the delay from the time that message is created, to the time that it is delivered to the desitnation node, as well as the portion of this delay that occurs at the source, and portion that occurs at the ferry.

Table 1: Node-Ferry contacts per hour

Ferry Model	Mobility Model for the Nodes: RWP	AB-RWP	CIRC
RWP	9.8	11.2	9.6
RWP-res	10.2	12.7	11.6
SFWP-3	8.4	8.0	8.8
SFWP-4	7.5	7.7	7.7
RCWP	14.2	13.0	10.10
OPWP	14.8	18.0	26.40

Table 1 shows the average node-ferry contacts per hour for each ferry routing scheme. OPWP is the clear winner for each node mobility model, except for the RWP node mobility model, where RCWP and OPWP give similar results (by chance, because the region center happens to be the optimal point for RWP motion).

Let's try to explain the performace of the different schemes. The SFWP schemes are clearly the worse performers. One might think that since SFWP schemes cover the entire region, they would perform well. However, there are good reasons for their poor performance. The length of the route (tour) for the SFWP-3 and SFWP-4 schemes is 39km and 32km respectively, compare this to the average route length of 9km that we observed for the OPWP for AB-RWP node mobility model. The longer route length means that the ferry takes a long time to complete one tour, and a significant fraction of this time is spent travelling in the parts of the region that have zero (parts that are not covered by smaller rectangles in fig. 4) or negligible probability of node presence. Note that even though the ferry covers the entire region, it does not cover the entire region at once, so the nodes and ferry can keep missing eachother.

The vanilla RWP ferry model suffers from somewhat the same problems as the SFWP models. The ferry may choose random way-points that are not in the region of any mobile node, thus time spent travelling to and from such way-points is completely futile, except for when the path to these way-points intersects the region of some mobile node. Even when the ferry does choose a way-point that is in the region of a mobile node, the probability that it would be at a point where mobile node is present is most likely to be is small. The reason that RWP performs better than the SFWP schemes is that the ferry tends to meet the nodes that are in the center of the region quite more often than the mobile nodes that are at the edges of region. This increases the average node-ferry meetings, but without any degree of fairness.

The RWP-res model alleviates part of the problems of vanilla RWP model by ignoring way-points that are not in the regions of the mobile nodes, and as a result, the RWP-res model performs slightly better than vanilla RWP. However, other problems of the RWP model remain.

The RCWP model does quite well when the nodes follow the RWP, or AB-RWP model. The reason is that RCWP uses a way-point that is the center of the region of the mobile node, and for the nodes moving according to the RWP model, the center also happens to be the point where the node is most likely to be present. Clearly, this advantage vanishes completely when nodes follow the CIRC mobility model.

Lastly, we found no significant instances of interference from node-node communication. In particular, there were only 0.03 contacts per hour per node, on average, where another node was found in the communication range of the serviced node or the message ferry when the nodes followed the RWP model, and the ferry followed the OPWP model. We found no such instances in our simulations when nodes followed AB-RWP or the CIRC mobility model.

Our second evaluation criteria is fairness. Figure 5 shows the node-ferry contacts per hour for individual nodes, and table 2 shows the fairness index for different ferry routing schemes, under different node mobility models. Clearly OPWP model has the least variation between the meetings for different nodes, and consequently yields an almost perfect fairness index value. An important point to note here is that since we allow a node to meet multiple times during a tour, it is possible that certain nodes may meet the ferry more frequently than other nodes and in that respect it can be argued that our method is not fair. However, our aim is to provide fairness in an approximately max-min sense. The ferry meets every node with certain minimum probability in each tour; any other potential contacts are akin to excess resources whose equal distribution we do not guarantee.[6] Our simulation results indicate that even without a explicit effort to distribute the "excess" contacts equally between the nodes, OPWP remains fair.

The higher variation between node-ferry contact frequency for the RWP ferry models deserves some attention. The RWP node visits the central region more often than the regions at the edges. This means that a RWP ferry would attempt to meet the nodes that are situated near the center of the region more often than the nodes that are situated farther away from the region, and as a result, the ferry would meet some nodes more often than others.

Finally, table 3 and figure 6 summarize the comparison based on delay and loss. We compute the end-to-end delay only for the messages that have been successfully downloaded; there still are messages in the ferry and source node buffer at the time of simulation termination, but we ignore these messages for the comparison. Note that while the

[6]In the strict max-min sense any excess resource is distributed equally between the customers; we do not guarantee such distribution.

Table 2: Fairness index for the ferry routing schemes

Ferry Model	Mobility Model for the Nodes: RWP	AB-RWP	CIRC
RWP	0.943	0.847	0.841
RWP-res	0.920	0.802	0.88
SFWP-3	0.832	0.864	0.894
SFWP-4	0.805	0.881	0.866
RCWP	0.943	0.919	0.565
OPWP	0.954	0.984	0.913

Table 3: Average delay and loss for different ferry routing schemes

Ferry Model	Avg. Delay(hr) E2E	Src	Ferry	Loss %
Node Mobility Model: RWP				
RWP	2.3	0.79	1.52	6.35%
RWP-res	2.78	0.98	1.80	7.52%
SFWP-3	2.50	0.77	1.73	6.38%
SFWP-4	3.69	1.43	2.28	19.81%
RCWP	2.34	0.80	1.53	3.4%
OPWP	2.25	0.73	1.52	0.77%
Node Mobility Model: AB-RWP				
RWP	4.41	1.10	2.63	13.9%
RWP-res	3.02	0.99	1.62	8.4%
SFWP-3	3.87	1.21	2.04	10.7%
SFWP-4	4.06	1.33	2.31	10.8%
RCWP	2.66	0.66	1.41	0.81%
OPWP	1.92	0.45	0.84	0%
Node Mobility Model: CIRC				
RWP	3.73	0.83	2.9	9.58%
RWP-res	3.61	0.79	2.82	6.32%
SFWP-3	4.16	0.85	3.31	5.79%
SFWP-4	4.23	0.72	3.51	9.64%
RCWP	2.81	1.0	1.81	24.8%
OPWP	1.69	0.26	1.43	0%

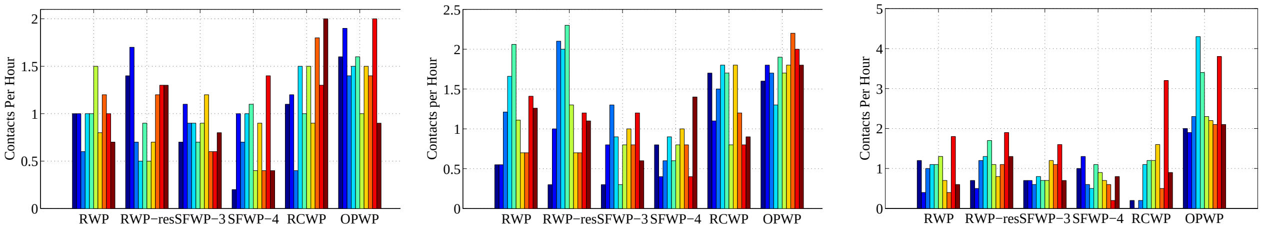
OPWP routing model does not do significantly better in terms of end-to-end delay for RWP node mobility model, it does better in terms of the message loss for this model. When the nodes follow the AB-RWP or the CIRC node mobility model, OPWP does clearly better both in terms of loss and end-to-end delay.

Overall, the OPWP model performs significantly better than any other model. The main reason is that we balance the travelling time and waiting time, and moreover invest that waiting time at way-points that are most advantageous in terms of increasing the contact probability with the nodes.

5.3 Performance Under Different System Parameters

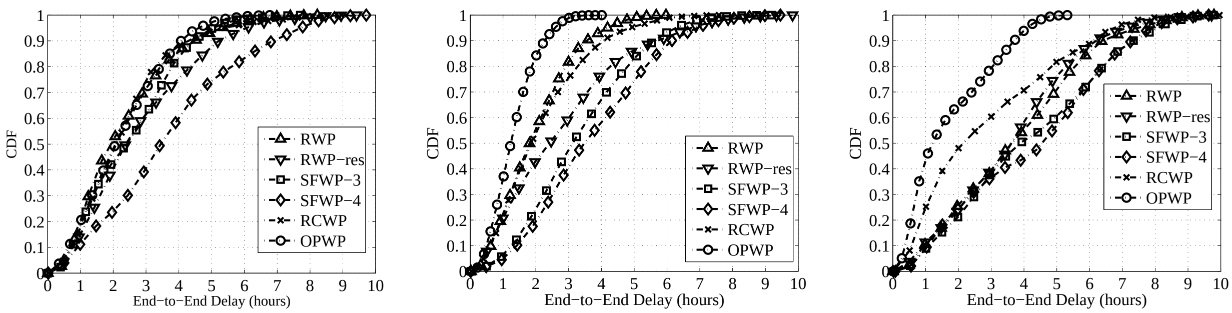
Performance of the message ferry routing schemes may depend on a number of system parameters. Here we evaluate the OPWP method under four parameters; (i) send and receive buffer size, (ii) service timers, (iii) sparsity of node deployment, and (iv) the speed of the ferry. There are other parameters of interest as well, such as, the radio range for communication, the bandwidth of the radio link, the message arrival and consumption rates at the nodes, and the speeds of the nodes, however, the above four are the most critical parameters, in our opinion.

Buffer size is a critical parameter for determining the message loss. We give a simplified explanation here. We dis-



(a) RWP Node Mobility Model (b) AB-RWP Node Mobility Model (c) CIRC Node Mobility Model

Figure 5: Node-Ferry meetings for individual mobile nodes under different ferry routing schemes, and different node mobility models. There is a bar in each group corresponding to each mobile node.



(a) RWP Node Mobility Model (b) AB-RWP Node Mobility Model (c) CIRC Node Mobility Model

Figure 6: CDF for End-to-End delay under different ferry routing and node mobility models.

Table 4: Loss rate under different buffer sizes with OPWP ferry routes

Node Model	Src & Dest. Buffer Size (no. of messages)			
	100	300	700	1000
RWP	64.51%	30.00%	3.89%	0.77%
AB-RWP	71.80%	20.00%	1.34%	0.0%
CIRC	43.13%	0.0%	0.0%	0.0%

cussed earlier that the node-ferry contact behaves as a geometric random variable. The expected number of messages that the node generates between successive contacts is $\lambda T/p$. If the node's source buffer can hold $k\lambda T/p$ messages, then the probability that there would be a loss is that there are no contacts for k/p consecutive tours; the probability of such a event reduces with increasing k, and the tail bounds can be used to bound these probabilities. In reality, the analysis is much more complex, because of residues that may be left from earlier service episodes [15]. Table 4 shows the loss rates for different node mobility models using different source and destination buffer sizes. Other parameters in the simulation are the same as described earlier in section 5.1.

The service timers also play a critical role in determining the loss rate. Recall from the service model in section 3 that if the *upload timer* expires before the source buffer is emptied, then there is a residue; if the upload timer is small, the residues will be larger and more ,requent, causing the source buffer to remain full, and increasing the delay and loss. Table 5 shows loss rates for different node mobility models when we use different source and destination buffer sizes. Other parameters in the simulation are the same as those that we described in section 5.1 above.

To analyze the system performance under different degrees of sparseness of node deployment, we consider setups where the 10 mobile nodes are deployed randomly in a 10km x 10km area, a 20km x 20km area, or a 40km x 40km area.

Table 5: Loss rate under different upload timers using OPWP ferry routes

Node Model	Upload Timer (seconds)					
	1	2	3	4	5	6
RWP	33.24%	9.34%	4.59%	3.69%	1.33%	0.77%
AB-RWP	35.63%	11.54%	4.30%	2.61%	0.4%	0%
CIRC	6.23%	0.0%	0.0%	0.0%	0.0%	0.0%

Table 6: Total contacts per hour for different sparseness levels - fainess index enclosed in paranthesis

Ferry Model	Area of Deployment		
	10km x 10km	20km x 20km	40km x 40km
OPWP	7.73 (0.961)	4.40 (0.977)	2.67 (0.976)
RCWP	5.53 (0.958)	1.47 (0.807)	1.33 (0.833)
RWP-res	2.60 (0.647)	0.87 (0.805)	0.67 (0.455)

We assume that the nodes are moving according to the Area Based RWP model that we had described earlier (each node continues to move in a 1km x 1km region). For each scenario we configure the nodes such that the source and destination buffers are twice the expected number of arrivals between contacts, and the service timers are such that they are sufficient to upload the expected number of messages that would arrive between contacts. Table 6 shows the total number of contacts per hour and the fairness index for different levels of sparseness. Note that the the contacts do decrease with sparseness, however, OPWP performs significantly better than other schemes.

Finally, we look at the impact of the ferry speed on the frequency of the node-ferry contacts. We keep the setup for the AB-RWP and CIRC node mobility models introduced in section 5.1. We assume that the ferry moves from one way-point to the next at a certain constant speed. Table 7, shows that the contact frequency increases with increas-

Table 7: Total contacts per hour for different ferry speeds with OPWP ferry routes

Node Model	Ferry Speed 5m/s	10m/s	20m/s
AB-RWP	13.7	18.8	23.1
CIRC	22.0	28.0	33.5

ing ferry speeds. However, we note that the increase is not linear. While increasing the ferry speed reduces the tour journey time linearly, its impact on the waiting time at the way-points depends on the node's mobility model. With the CIRC mobility model, the waiting time only depends on the node speed, and not on the ferry speed. In the case of AB-RWP mobility, increasing the ferry speed decreases T_j, which results in lower optimal contact probability, and consequently also lower the waiting times, but the relationship is not linear with the speed of the ferry.

6. CONCLUSIONS AND FUTURE WORK

We have described a new framework for designing routes for message ferries that does not require any online collaboration between the nodes and the ferry. The ferry route comprises a carefully chosen ordered set of way-points and waiting times at these way-points, such that each time that the ferry traverses this route, it contacts every node with a certain minimum probability. We have demonstrated through extensive simulations that our scheme performs significantly better than other naive ferry routing schemes. In the process we have gained valuable insight regarding how the information from mobility models can be used to design better routes.

In the future, we would like to consider the case of using multiple message ferries, as these would be needed in environments which are too sparse for a single ferry, or where the traffic demands are too high. We also plan to futher elaborate and develop the enhancements that we have mentioned in section 4.5.

7. REFERENCES

[1] A. Beaufour, M. Leopold, P. Bonnet. Smart-tag Based Data Dissemination. In proc. First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA). Sep. 2002.

[2] S. Boyd, L. Vandenberghe. Convex Optimization. Cambridge University Press. Mar. 2004.

[3] Z. Chen, H. Kung, and D. Vlah. Ad hoc Relay Wireless Networks Over Moving Vehicles on Highways. In proc. The 2001 ACM Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc'2001). Oct. 2001.

[4] W. Cook, D. Applegate, R. Bixby, and V. Chvatal. Concorde: A code for solution of Travelling Salesman Problem. http://www.tsp.gatech.edu/

[5] J. Davis, A. Fagg, and B. Levine. Wearable Computers as Packet Transport Mechanisms in Highly-partitioned Ad hoc Networks. In proc. IEEE International Symposium on Wearable Computing. Oct. 2001.

[6] K. Fall. A Delay-Tolerant Network for Challenged Internets. In proc. ACM SIGCOMM 2003. Aug. 2003.

[7] Y. Gu, D. Bozdag, E. Ekici, F. Ozguner, and C. Lee. Partitioning Based Mobile Element Scheduling in Wireless Sensor Networks. In. proc. Second Annual IEEE Conference on Sensor and Ad Hoc Communications and Networks (SECON) 2005.

[8] R. Jain, A. Durresi, G. Babic. Throughput Fairness Index: An Explanation. ATM Forum/99-0045, Feb. 1999. http://www.cse.wustl.edu/ jain/atmf/a99-0045.htm

[9] R. Jain, D. Lelescu, and M. Balakrishan. An Empirical Model for User Registration Patterns in a Campus Wireless LAN. In the proc. Tenth Annual International Conference on Mobile Computing and Networking (Mobicom 2005). Sep. 2005.

[10] S. Jain, K. Fall, R. Patra. Routing in Delay Tolerant Networks. In proc. ACM SIGCOMM 2003.

[11] S. Jain, M. Demmer, R. Patra, and K. Fall. Using Redundancy to Cope with Failures in a Delay Tolerant Network. In proc. ACM SIGCOMM 2005.

[12] D. Johnson and D. Maltz. Dynamic Source Routing in Ad hoc Wireless Networks. In proc. ACM SIGCOMM. 1996.

[13] A. Kansal, A. Somasundara, D. Jea, M. Srivastava, D. Estrin. Intelligent Fluid Infrastructure for Embedded Networking. In proc. Second International Conference on Mobile Systems, Applications and Services (Mobisys 2004). Jun. 2004.

[14] J. Leguay, T. Friedman and V. Conan. DTN Routing in a Mobility Pattern Space. In proc. ACM SIGCOMM 05 Workshop on Delay Tolerant Networking and Related Topics (WDTN-05) 2005.

[15] K. K. Leung. and M. Eisenberg. A Single-Server Queue with Vacations and Gated Time-Limited Service. IEEE Transactions on Communication. Vol. 38. No. 9. Sept. 1990.

[16] Q. Li, and D. Rus. Sending Messages to Mobile Users in Disconnected Ad hoc Wireless Networks. In. proc. 4th ACM/IEEE Internation Conference on Mobile Computing and Networking (Mobicom'98). Nov. 1998.

[17] J. Luo, and J. Hubaux. Joint Mobility and Routing for Lifetime Elongation in Wireless Sensor Networks. In. proc. IEEE Infocom, 2005.

[18] S. Merugu, M. Ammar and E. Zegura. Routing in Space and Time in Networks with Predictable Mobility, Technical report: GIT-CC-04-07. College of Computing, Georgia Institute of Technology. 2004.

[19] C. Perkins and P. Bhagwat. Highly Dynamic Destination-Sequenced Distance-vector Routing (DSDV) for mobile computers. Computer Communications Review. 24. Oct. 1994.

[20] C. Perkins, and E. Belding-Royer. Ad hoc On-demand Distance Vector Routing. In proc. 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'99). Feb. 1999.

[21] R. Shah, S. Roy, S. Jain, and W. Brunette. Data MULEs: Modeling a Three-tier Architecture for Sparse Sensor Networks. Elsevier Ad Hoc Networks Journal, vol. 1, issues 2-3, Sept. 2003, pp. 215-233.

[22] A. Somasundara, A. Ramamoorthy, and M. Srivastava. Mobile Element Scheduling for Efficient Data Collection in Wireless Sensor Networks with Dynamic Deadlines. In. proc. 25th IEEE International Real Time Systems Symposium (RTSS'04), 2004.

[23] T. Small, Z. Haas. The Shared Wireless Infostation Model – A New Ad Hoc Networking Paradigm (or Where there is a Whale, there is a Way). In proc. The Fourth ACM International Symposium on Mobile Ad hoc Networking and Computing. (Mobihoc 2003). June, 2003.

[24] T. Small and Z. Haas, Resource and Performance Tradeoffs in Delay-Tolerant Wireless Networks. In proc. ACM SIGCOMM 05 Workshop on Delay Tolerant Networking and Related Topics (WDTN-05) August 2005.

[25] M. Tariq. Interarrival Times for the Random Waypoint Model are Exponentially Distributed. http://www.cc.gatech.edu/ mtariq/pub/notes/rwp-interarrivals.pdf

[26] A. Vahdat and D. Becker. Epidemic Routing for Partically-Connected Ad hoc Networks. Technical Report. Duke University. 2000.

[27] V. Vazirani. Approximation Algorithms. Springer. Aug. 2001.

[28] J. Yoon, M. Liu, B. Noble. Random Waypoint Considered Harmful. In. proc. IEEE Infocom 2003, San Franciso, CA.

[29] W. Zhao, M. Ammar, and E. Zegura. Proactive Routing in Highly-partitioned Wireless and Ad hoc Networks. In proc. 9th IEEE Workshop on Future Trends in Distributed Computing Systems (FTDCS). May, 2003.

[30] W. Zhao, M. Ammar, and E. Zegura. A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks. In proc. The 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'2004). May, 2004.

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