LIPIcs Leibniz International Proceedings in Informatics

Algorithms for Message Ferrying on Mobile ad hoc Networks

Mostafa Ammar[1], Deeparnab Chakrabarty[2], Atish Das Sarma[1], Subrahmanyam Kalyanasundaram[1], Richard J. Lipton[1]

[1] Georgia Institute of Technology
{ammar, atish, subruk, rjl}@cc.gatech.edu

[2] University of Waterloo
deepc@math.uwaterloo.ca

ABSTRACT. Message Ferrying is a mobility assisted technique for working around the disconnectedness and sparsity of Mobile ad hoc networks. One of the important questions which arise in this context is to determine the routing of the ferry, so as to minimize the buffers used to store data at the nodes in the network. We introduce a simple model to capture the ferry routing problem. We characterize *stable* solutions of the system and provide efficient approximation algorithms for the MIN-MAX BUFFER PROBLEM for the case when the nodes are on hierarchically separated metric spaces.

1 Introduction

Message Ferrying is a new approach developed to assist communication in Mobile ad-hoc networks [6, 15, 16, 17, 18]. Mobile ad-hoc networks are typically deployed with limited infrastructure. Moreover, due to various conditions like limited radio range, physical obstacles or inclement weather, some nodes in the network might not be able to communicate with others. This could result in a disconnected network. In such situations, a typical network protocol might not yield good results. Message Ferrying is an approach which works around such problems. The message ferrying technique makes use of mobile nodes, called "ferries", which are able to collect and transport data from one node to another. Message ferries move around the deployed area according to known routes and communicate with other nodes they meet. By using ferries as relays, nodes can communicate asynchronously with other nodes that are disconnected.

The Message Ferrying scheme raises many theoretical questions that are currently open. For example, Zhao et.al [17] have developed *ad hoc* codes that decide how the ferries should move. While these codes appear to perform well in simulations there are no bounds on the performance of their heuristic methods. The data at the nodes has to be locally stored in buffers till it can be passed on to the ferries. In this paper, we look at the buffer optimization problem for the nodes in the network. We devise routing schemes for the ferries so that the maximum buffer utilization at any node is minimized.

We do the following in this paper.

- Formalize models for the Message Ferry routing problems.
- State exact conditions for the *Stability Problem* of ensuring finite buffers.

Foundations of Software Technology and Theoretical Computer Science (Kanpur) 2009.
Editors: Ravi Kannan and K. Narayan Kumar; pp 13–24
Leibniz International Proceedings in Informatics (LIPIcs), Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Germany.
Digital Object Identifier: 10.4230/LIPIcs.FSTTCS.2009.2303

- Devise approximation algorithms for buffer optimization.

We feel that the main contribution of this work is in the creation of the precise models for the Message Ferrying problem. We feel that similarly motivated problems, stemming from direct practical applications could be important in the future. The algorithms and techniques in this paper are fairly preliminary; these are just the first steps towards understanding the Message Ferrying problem.

1.1 Model and Problem Statement

We model the problem in the following manner. We look at all the connected components of the network. Since the network can communicate within a connected component using traditional protocols, we can model each connected component as a node. The connected components are modeled as nodes numbered $[n]$ in a metric space. The metric space induces a distance $d(\cdot,\cdot)$ on the nodes. The ferries are assumed to be devices with infinite storage capacity traveling across the space at unit speed. We assume that node i generates data at a rate r_i which is to be passed on to the ferries (the data can then in turn be transferred from the ferry to other nodes). The rate of data transfer from the nodes to the ferries is given by r_F. Unless otherwise stated, we shall assume that there is only one ferry. We wish to provide message ferrying schemes which are *optimal*.

A Message Ferrying scheme is an (infinite) sequence σ of tuples (i,t), where i denotes the node visited and t the time spent exchanging data with i in this visit. We need to look at the case where the ferry can read data at an infinite speed, i.e. $r_F = \infty$, separately. In this case, it is clearly undesirable to spend any time at a node, as all the data is read instantaneously. Then all we need to specify is a sequence of nodes. Although the sequence could be infinite, we prove that periodically repeating sequences of finite periods suffice for our purposes.

Any viable Message Ferrying scheme would need to be optimized over a large number of disparate parameters like delay minimization and packet loss. Currently, ad hoc and complex measures are used for performance evaluation [7, 18]. We propose two concrete measures and present results for the same. Our first measure is the notion of *stability*. A Message Ferrying scheme is called stable, if the maximum amount of data stored at a node at any point of time is bounded. This is clearly desirable as a node can have only a fixed finite buffer.

DEFINITION 1.STABILITY PROBLEM
Given the rates r_i and r_F, and the distance metric d, is there a Message Ferrying scheme such that the required buffers at all the nodes are bounded?

Our second problem is to optimize the maximum buffer size required over all nodes. That is, given that the rates satisfy the stability conditions, we need to find the scheme minimizing the maximum buffer of any node. Later, we shall see (theorem 20) that the stability criterion for the problem with multiple ferries reduces to that with a single ferry. So it makes sense to look at the optimization problem when only one ferry is involved.

DEFINITION 2.MIN-MAX BUFFER PROBLEM (MMBP)
Given the data rates r_i and ferry rate r_F, and the distance metric d, find the order of ferry visits so as to minimize the maximum buffer required at any node.

As we shall see in Section 3, the above problem turns out to be as hard as solving TSP on the same metric. The MMBP problem is thus a variant of the TSP problem where the objective is not to optimize the length of the tour, rather minimize a sort of weighted delay. Variants of TSP [1, 13], have been studied extensively in the past, although none imply anything about the problem we study.

In section 2, we characterize the necessary and sufficient conditions for stability of a solution. In section 3, we first prove the NP hardness of MMBP problem. We subsequently obtain approximation algorithms for MMBP under restricted metric spaces. In particular, we give a constant factor approximation algorithm for *hierarchically well separated trees* (HSTs) of constant height, and, a $\frac{4}{3}$-factor approximation ratio for the uniform metric case. We extend this to an $O(n)$-factor algorithm for HSTs of height $O(\log n)$. Notice that even though n may be large, this approximation ratio holds for arbitrarily large rates r_i of data generation at nodes as well. We look at some simple extensions to the MMBP problem in section 4. In section 5, we conclude with some remarks and open problems suggesting specific future directions of work.

2 Characterization of stable instances

In this section, we give necessary and sufficient conditions for the existence of stable solutions. We consider the case when there is only one ferry, the node data rates are r_i, ferry rate is r_F and the data is only sent from nodes to the ferry. We then use this to obtain results for the general case.

For the case mentioned above, note that $\sum_{i=1}^{n} r_i < r_F$ is a necessary condition for a stable solution if any of the distances are non-zero. Otherwise the total rate of generation of data in the nodes exceeds the rate at which it can be read by the ferry. In the next theorem, we show that this necessary condition is also sufficient.

THEOREM 3. *Given rates r_i for the nodes, ferry rate r_F, any distances d, a stable solution exists if $\sum r_i < r_F$.*

PROOF. Consider the sequence that visits nodes in order 1 to n, spending time $t_1, t_2, \ldots, t_n$ at them respectively and repeats. Suppose it takes time T to travel from 1 to n in that order.

Now notice that the following is enough for stability: for every node j, the amount of data consumed by the ferry in one visit must be at least the amount that is generated between two visits of the ferry to the node j. That is, for every j we have

$$r_F \cdot t_j \geq r_j \cdot (T + \sum_{i=1}^{n} t_i)$$

Adding these equations over all j we get

$$r_F \cdot \sum_{j=1}^{n} t_j \geq (\sum_{j=1}^{n} r_j) \cdot (T + \sum_{i=1}^{n} t_i)$$

When $r_F = \sum_{j=1}^{n} r_j + \varepsilon$, for some $\varepsilon > 0$, it is easy to check that $t_i = \frac{T}{\varepsilon} r_i$ satisfies the inequalities, implying a stable solution. ∎

3 Min-max Buffer Problem (MMBP)

In this section, we look at the general min-max buffer problem. Throughout this section, we assume that the instance is stable. Also, hereafter, we would be assuming that there is only one ferry node. A solution is called *periodic* if the nodes are visited in a periodic pattern (note that this pattern could have some nodes occurring more than once). In the following proposition, we show that we could look for periodic optimal solutions, since they are as good as optimal solutions, in case of rational rates.

PROPOSITION 4. *For any instance of the* MMBP *with rational rates and distances, there exists a periodic Message Ferrying scheme which is optimal.*

PROOF. Suppose there is an optimal aperiodic solution with maximum buffer B. By hypothesis, the rates and distances are rational. This implies that the optimal solution is rational, and when the ferry reaches a node, the buffer state is rational. Scaling the states to be integral, and recalculating B, each buffer can be one of $0, 1, 2, \ldots, B$ at any given point of time. If there are n nodes, there can be at most $(B+1)^n$ possible buffer states. There are n nodes, we can consider a combined notion of states $S = (\bar{B}, i)$, where $\bar{B}$ is a vector denoting the buffer state across all nodes and i denotes the node visited. So the optimal aperiodic solution returns to at least one of these states S more than once. Let us say that the repeated state is $S^* = (\bar{B}^*, i^*)$. Consider a new periodic solution where this subsequence (between two repetitions of S^*) is repeated indefinitely. Since the same visits are conducted between the two visits to S^*, upon returning to i^*, the buffers have again come back to $\bar{B}^*$. Since the buffers never overflowed in the original aperiodic sequence, they do not overflow in this repeated sequence. This is because we go through the states which were all part of the original aperiodic sequence. Thus we have a periodic sequence which is optimal. ∎

Henceforth, our solutions will be a sequence that is repeated periodically. The following proposition shows the relation of MMBP to the TSP if all the data production rates are identical. Note that this is not true in general. When the rates are different, the solution given by the TSP can be arbitrarily bad for the message ferrying problem.

PROPOSITION 5. *For any underlying metric $d(\cdot,\cdot)$, if the rates of all nodes are equal, i.e., $r_i = 1$, for all i, and the rate of the ferry $r_F = \infty$, then finding the optimal solution to the* MMBP *is the sequence generated by the optimal Traveling Salesman Problem (TSP) tour, and hence NP-hard.*

PROOF. Recall that since $r_F = \infty$, we need to only specify the routing order, there is no need to mention wait times at each node. It is enough to show that the optimal sequence for the MMBP must be generated by a tour. Since all rates are the same, the maximum buffer for the sequence generated by a tour is proportional to the cost of the tour. This implies that the optimal ferry route for the MMBP is the optimal TSP tour.

Assume that the optimal sequence is not a tour. Let σ be the sequence of the optimal solution. Let us relabel the nodes according to the order that we see them in the optimal solution. By the choice of the labeling, the last node to be visited is n. Since σ is not the optimal TSP tour, n will be seen for the first time after the ferry has traveled a distance greater than the cost of the optimal TSP tour. Since every node has the same rate, the buffer of n is the largest of all the nodes till we visit n for the first time. Consider a solution τ, where the ferry visits all the nodes repeatedly as in the optimal TSP solution. The maximum buffer of n in τ is less than the buffer of n in σ. Moreover, the maximum buffer of every node in τ is the same. Thus τ has a strictly lesser maximum buffer than the route σ. ∎

The above proposition states that solving MMBP is at least as hard as solving TSP on the same underlying metric. Papadimitriou and Yannakakis in [14] prove that the TSP is NP hard even when the distances of the graph are restricted to 1 and 2. This implies that the MMBP is NP hard, even for the case when the distances are restricted to 1 and 2.

In the next two sections, we investigate approximation algorithms when the rate of the ferry is infinite.

3.1 Uniform Metric Case

Here we look at the uniform metric case, where the distance between all nodes are the same. That is, $d(i, j) = 1$ for $i \neq j$. The nodes have rates r_i and we have one ferry, with rate $r_F = \infty$. Once again, recall that since the ferry rate is infinite, we just need to mention the next node to be visited and there is no need to specify wait times. For this case we prove the following theorem.

THEOREM 6. *There is a $\frac{4}{3}$-factor approximation algorithm for* MIN-MAX BUFFER PROBLEM *in the case when the metric is uniform, and the ferry rate $r_F = \infty$.*

The algorithm outline is as follows. Given a guess of the max-buffer B, the algorithm checks *approximate* feasibility of B. That is, the algorithm rejects B only if it is infeasible, otherwise it returns a solution with a max-buffer guarantee of $\frac{4}{3}B$. The $\frac{4}{3}$-approximation follows from a binary search on the possible values of B.

Let σ be any (infinite) feasible sequence of the node visits with max-buffer B. Each node $i \in [n]$ must be visited once in every B/r_i steps. If $d(i)$ denotes the maximum distance between two consecutive appearances of i in σ, we must have $d(i) \leq \lfloor B/r_i \rfloor$. Hence the feasibility solution for the uniform metric case reduces to the following combinatorial problem, called the *pinwheel scheduling problem*. Let us set $m_i = \lfloor B/r_i \rfloor$.

> PINWHEEL SCHEDULING PROBLEM:
> Given integers $m_1 \leq \cdots \leq m_n$, is there a (infinite) sequence σ of $[n]$, such that, for each $1 \leq i \leq n$, the maximum distance between any two consecutive appearances of i is at most m_i? If it does, we call $(m_1, m_2, \cdots, m_n)$ feasible.

This scheduling problem is of independent interest and has been studied previously [3, 4, 9, 10, 11, 12]. Here are some observations about this problem.

PROPOSITION 7. *An instance of the pinwheel scheduling problem $(m_1, m_2, \cdots, m_n)$ has a feasible solution, only if $\sum_{i=1}^{n} \frac{1}{m_i} \leq 1$.*

PROOF. Consider a snapshot of any feasible σ of length $Z = m_1 m_2 \cdots m_n$. For each i, It must contain at least Z/m_i occurrences of i. Since there are only Z possible slots, we have $Z \geq \sum_{i=1}^{n} Z/m_i$ proving the lemma. ∎

Remarks: Notice that this condition is necessary but not sufficient; consider $(m_1, m_2, m_3) = (2, 3, N)$. In this case, $\sum \frac{1}{m_i}$ is approximately $\frac{5}{6}$ for large N, but there is no sequence that can satisfy this for any finite N.

Let OPT be the optimal maximum buffer value, amongst all feasible routes of the ferry. Notice that the optimal routing solves the sequence feasibility problem for the rates $(OPT/r_1, OPT/r_2, \cdots, OPT/r_n)$. So this sequence is feasible, and so proposition 7 implies the following lemma.

LEMMA 8. *If the nodes have rates $r_1, r_2, \ldots, r_n$, with uniform metric, and $r_F = \infty$, we have*

$$OPT \geq \sum_{i=1}^{n} r_i$$

where OPT is the optimal maximum buffer value.

We now have a direct reduction from the Pinwheel Scheduling problem to the ferry routing problem.

LEMMA 9. *Let $\alpha \geq 1$. If we have an algorithm for the pinwheel scheduling problem for m_i such that $\sum_{i=1}^{n} \frac{1}{m_i} \leq \frac{1}{\alpha}$, then we have an α-approximation algorithm for the MMBP problem, with uniform metric, and $r_F = \infty$.*

PROOF. Given a target buffer B, let $m_i = \lfloor \alpha B / r_i \rfloor$. Note that m_i is the maximum allowed time gap between any two consecutive visits to the node i, if we want to bound the buffer by αB. If $\sum_{i=1}^{n} \frac{1}{m_i} > \frac{1}{\alpha}$, then $\sum_{i=1}^{n} \lfloor B/r_i \rfloor^{-1} > 1$. This by Lemma 7 implies that B is infeasible, and the algorithm rejects it. If not, then the algorithm for pinwheel scheduling returns a feasible sequence for $(m_1, m_2, \cdots, m_n)$. For this sequence, the maximum buffer of any node is at most αB. Thus this is an α-approximation.

The only remaining decision is the choice of B. By lemma 8, we have $OPT \geq \sum_{i=1}^{n} r_i$. Also, we can see that $OPT \leq B_{\max}$ where $B_{\max} = \alpha \sum_{i=1}^{n} r_i + r_{\max}$. This is because if we set $B = B_{\max}$, then the corresponding value $\sum_{i=1}^{n} \frac{1}{m_i} \leq \frac{1}{\alpha}$. So in order to complete the approximation algorithm, we need to do a binary search for B between $\sum_{i=1}^{n} r_i$ to $\alpha \sum_{i=1}^{n} r_i + r_{\max}$. ∎

We can use the above lemma 9, with an approximation algorithm for pinwheel scheduling. Fishburn and Lagarias in [9] gave an algorithm for pinwheel scheduling as long as the following condition is met.

THEOREM 10.[Fishburn, Lagarias] *There exists an algorithm for the pinwheel scheduling problem when $\sum_{i=1}^{n} \frac{1}{m_i} \leq 0.75$.*

Theorem 10 along with lemma 9 gives us the following approximation algorithm.

THEOREM 11. *There is a $\frac{4}{3}$-factor approximation algorithm for the* MIN-MAX BUFFER PROBLEM *in the case when the metric is uniform, and the ferry rate $r_F = \infty$.*

The above theorem straightaway implies a $\frac{4}{3}\frac{D_{max}}{D_{min}}$ factor for general metrics where D_{max} (D_{min}) is the maximum (minimum) distance between two points. In particular, in the case where the distances are 1 and 2, this implies a $\frac{8}{3}$ factor approximation. Note that by Proposition 5 and the paper [14], this instance is already NP-hard.

Theorem 10, together with lemma 9 also implies the following lemma, which is used in the section 3.2.

LEMMA 12. *Given nodes of rate $r_1, \cdots, r_n$ and a distance 1 between each node, there exists a ferry routing with maximum buffer at most $\frac{4}{3}\sum r_i + r_{max}$.*

A Simpler Algorithm for Pinwheel Scheduling

Fishburn and Lagarias' algorithm for the pinwheel scheduling problem is quite involved. The algorithm involves case based analysis for several small sets of problem instances, classifies the small sets and extends it to bigger sets based on the classification. We therefore now give a simpler algorithm for pinwheel scheduling, with a slightly worse bound. Our algorithm works when $\sum_{i=1}^{n}\frac{1}{m_i} \le 1/2$.

LEMMA 13. *If $\sum_{i=1}^{n}\frac{1}{m_i} \le 1/2$, then $(m_1, m_2, \cdots, m_n)$ is feasible for pinwheel scheduling.*

PROOF. We prove by induction on n. The base case of $n = 1, m_1 = 2$ is trivial. For $n \ge 2$, we have $\sum_{i=1}^{n}\frac{1}{m_i} \le 1/2$. Rearranging and dividing we get $\sum_{i=2}^{n}\frac{1}{\bar{m}_2} \le 1/2$, where

$$\bar{m}_i = \left\lceil 2m_i(\frac{1}{2} - \frac{1}{m_1}) \right\rceil$$

By induction, we get $(\bar{m}_2, \cdots, \bar{m}_n)$ is feasible. Let σ' be the feasible pinwheel scheduling sequence. Obtain σ by putting 1 in σ' every m_1 positions. This increases the distance between two i's to $\bar{m}_i + \left\lceil \frac{\bar{m}_i}{m_1 - 1} \right\rceil$. We would have a feasible sequence for $(m_1, m_2, \cdots, m_n)$ if $\bar{m}_i + \left\lceil \frac{\bar{m}_i}{m_1-1} \right\rceil \le m_i$. The following claim shows that this is indeed the case. ∎

CLAIM 14. *For any integers $1 \le m_1 \le m_i$ let $\bar{m}_i = \left\lceil m_i - \frac{2m_i}{m_1} \right\rceil$, we have that*

$$\bar{m}_i + \left\lceil \frac{\bar{m}_i}{m_1 - 1} \right\rceil \le m_i$$

PROOF. Let $x = \frac{2m_i}{m_1}$ and let $k = \lfloor x \rfloor$. Note $\bar{m}_i = m_i - k$. Thus to prove the claim, it suffices to show $\left\lceil \frac{m_i - k}{m_1 - 1} \right\rceil \le k$. Since $m_i - k = m_1 x/2 - k$, we have

$$\left\lceil \frac{m_i - k}{m_1 - 1} \right\rceil = \left\lceil \frac{x}{2} + \frac{1}{m_1 - 1}(\frac{x}{2} - k) \right\rceil \le k$$

because $\frac{x}{2} < \lfloor x \rfloor = k$, for all $x > 1$. ∎

This sufficiency condition is constructive as well. Given $(m_1, m_2, \cdots, m_n)$ such that $\sum_{i=1}^{n} 1/m_i \leq 1/2$, recursively run on $(\bar{m}_2, \bar{m}_3, \cdots, \bar{m}_n)$ and put 1 every m_1 spots in the sequence returned.

In the section 3.2, we give a constant factor algorithm for the metrics induced by hierarchically separated trees of constant depth.

3.2 Metrics induced by HSTs of constant depth

In this section, we generalize the results of the previous section to a greater class of metrics. We have seen that we can get a constant factor approximation algorithm for the uniform metric. In this section, we shall look at metrics induced by hierarchically well separated trees (HST).

DEFINITION 15. *A* Hierarchically well Separated Tree *(HST) is a rooted tree such that any pair of leaves that have the least common ancestor at height i, are separated by a distance of D^{i-1}, where D is a parameter.*

HSTs induce a metric on the nodes of the tree. These metrics have been widely studied [2, 5, 8] in the area of metric embeddings. HSTs are interesting because it is possible to get low-distortion embeddings of general metrics into those induced by HSTs.

In this section, we show a constant factor approximation for metrics induced by HSTs of constant height. Note that the uniform case is an HST of height 1. For the sake of clarity, we first look at the case of height 2. We call these metrics $\{1, D\}$-metrics. (Note that any metric with distances only 1 and D, with $D > 2$, can be thought of as an $\{1, D\}$-metric).

In this case, we can partition the point sets into clusters $P_1, P_2, \cdots, P_t$ with each pair in any cluster being at distance 1, and any two points in different clusters at distance D.

We fix some notation. Let $R_i := \max_{j \in P_i} r_j$ and $S_i := \sum_{j \in P_i} r_j$ be the maximum rate and sum of rates for nodes in P_i. Our algorithm would maintain $B_1, \cdots, B_t$ as the max buffers needed for the various clusters. Note that the max-buffer $B = \max_i B_i$.

We now state lower bounds on OPT for this instance.

LEMMA 16.
1. $OPT \geq \sum_{i=1}^{t} S_i$
2. $OPT \geq \sum_{i=1}^{t} DR_i$

PROOF. Note that if we shrink distances between the nodes and or delete nodes, we could only decrease the optimum buffer value. If the distance D were shrunk to 1, then by Lemma 8 we get $OPT \geq \sum_i r_i = \sum_i S_i$. If we delete all points other than the ones with maximum rate, again by Lemma 8 (recall that we are on an instance where distances are scaled by a factor D), we get $OPT/D \geq \sum_i R_i$. ∎

Suppose σ_i be the sequence corresponding to Lemma 12 in Section 3.1 for the nodes in P_i. This guarantees a max-buffer of $\frac{4}{3}S_i + R_i$. We use this fact to develop the following algorithm for the $\{1, D\}$ case. We run the uniform metric algorithm at two separate levels for this case.

Algorithm $\{1, D\}$

1. Visit each cluster P_i once in every window of k_i clusters visited. (k_i will be determined later)
2. When at P_i, run D steps of σ_i starting from the point where it was when it last left P_i.

What this algorithm essentially does is to spend *some* time in each cluster of nodes (here a cluster is a set of nodes with pairwise distances 1). Across the clusters, the algorithm simulates the uniform metric algorithm, where the rate of data generation of a cluster is simply the sum of rates of data generation of the nodes in the cluster. Further, whenever the algorithm spends time inside a given cluster, the algorithm again simulates the uniform metric algorithm within it. Notice, however, that the algorithm may not necessarily be able to perform an entire loop over all nodes within a cluster in one visit. Therefore, it resumes the optimal algorithm within the cluster once it returns to the cluster the next time.

THEOREM 17. *The above algorithm achieves a constant factor approximation for the* MIN-MAX BUFFER PROBLEM *on* $\{1, D\}$ *metrics.*

PROOF. We argue cluster by cluster. We have two cases.
Case 1: Every node in P_i is visited after at least D steps in σ_i. Since for each D steps of σ_i, the algorithm spends $2k_iD$ time outside P_i (k_iD for traveling across clusters and D in each of the k_i clusters), the time between two consecutive occurrences of the point is increased by a factor of at most $2k_iD/D = 2k_i$. Thus we have $B_i \leq 2k_i(\frac{4}{3}S_i + R_i) \leq \frac{14}{3}k_iS_i$ from the lemma 12.
Case 2: There is a node in P_i which is visited with a gap strictly less than D in σ_i. This implies that it is visited every time we visit the cluster, and thus its buffer is at most $R_i(2k_iD)$.

The two cases give $B_i \leq \max(\frac{14}{3}k_iS_i, 2R_ik_iD) \leq (\frac{14}{3}S_i + 2R_iD)k_i$. By choosing $k_i = \left\lceil \frac{\frac{4}{3}\sum_{i=1}^{r}(2R_iD + \frac{14}{3}S_i)}{2R_iD + \frac{14}{3}S_i} \right\rceil$, we get $B_i \leq \frac{4}{3}\sum_{i=1}^{r}(2R_iD + \frac{14}{3}S_i) \leq 9 \cdot OPT$ from Lemma 16.

We complete the proof by noting that a visiting sequence for the clusters for these k_i's can be achieved since $\sum_{i=1}^{t} 1/k_i \leq 0.75$ and we are done by Lemma 8. ∎

THEOREM 18. *There is a constant factor approximation to the MMBP for HSTs of constant height.*

PROOF. Assume that we have a C-factor approximation algorithm to an HST of height k; let this algorithm be A_C. Now consider an HST of height $(k+1)$, say it has t subtrees of height k. We claim that the analogous extension to $Algorithm\{1, D\}$ works here:

1. Visit the points in subtree i, P_i, in every window of k_i subtrees visited among the t subtrees.
2. When at P_i, run D^k steps of algorithm A_C from the point where it was when it last left P_i.

Running through a similar analysis as in $Algorithm\{1, D\}$, we see that with an increase of 1 in the height of the HST, the approximation ratio increases by a factor at most 7. Thus we get an approximation factor of $\frac{4}{3} \cdot 7^k$ where the height of the HST is k. ∎

Remarks: By the above algorithm, we achieve an $O(n)$-factor approximation ratio on an HST of height $O(\log n)$.

Remarks: Standard metric embedding results of Bartal [2] and Fakcheroenphol et al. [8] give a probabilistic embedding of HSTs of height $\log n$ into arbitrary metrics with an expected distortion of $O(\log n)$. This has been used in obtaining approximation algorithms for various problems like the buy-at-bulk network design, metric labeling, etc. which solve the problem on HST instances and extend it to general metrics via results of [8].

Unfortunately these results do not help us guarantee any approximation on the MMBP problem for general metrics as we deal with *maximum* buffers and the expectation of the maximum buffer can be much larger than the maximum of the expected buffer sizes.

4 Extensions

In this section, we show simple extensions of the stability conditions for data exchange, and for the case when there are multiple ferry nodes collecting data.

4.1 Data Exchange Problem

Suppose a node i generates data at a rate a_i and the ferry generates data, to be passed on to the node i, at a rate b_i. The following lemma follows easily from Theorem 3.

LEMMA 19. *If the ferry can only receive or send data at a time, a stable Message Ferrying scheme exists if and only if $\sum a_i + b_i < r_F$. If the ferry can receive and send simultaneously, a stable Message Ferrying scheme exists if and only if $\sum \max(a_i, b_i) < r_F$.*

Note that $\max(\sum a_i, \sum b_i) < r_F$ is not a sufficient condition for the simultaneous case of the above lemma. A simple example is two nodes, with $a_1 = b_2 = 0.8, a_2 = b_1 = 0.1, r_F = 1$.

Consider the stability problem in a situation where the ferry (or ferries) have bounded buffers. Suppose each of the ferry has a limited buffer size. Notice that in such a case, we cannot get a similar theorem like theorem 3. When one limits the ferry buffer size, the stability is not just a function of the rates of the nodes. The stability would depend on the topology of the nodes as well. To see this, consider the following problem: there is one ferry and two nodes, and the ferry has to transport data from one node to the other. The rates r_1, r_2 correspond to the transfer rates at the nodes. Let $r_1 + r_2$ be infinitesimally close but still less than r_F. By theorem 3, we would still be stable if the ferry had no bounds on its buffer. Theorem 3 achieves this by making the ferry stay very long at either node, and then moving only occasionally. But a bound of the ferry buffer size would force the ferry to move earlier, and thus risk losing data or being not stable. Thus a bound on the ferry's buffer size would mean that the stability depends on the topology of the problem.

4.2 Multiple Ferries

Suppose there are m ferries each with the same ferry transfer rate, r_F. Also assume that at any node only one ferry can operate at a time. We have the following necessary and sufficient conditions for this case.

THEOREM 20. *Consider the case when there are m ferries each with the same ferry transfer rate, r_F. Also, at any node only one ferry can operate at a time. The necessary and sufficient condition for stability is $\sum r_i < mr_F$ and $r_i \leq r_F$ for all i.*

PROOF. One of the necessary conditions, $\sum r_i < mr_F$ is immediate. If $r_i > r_F$ for any node, notice that at most one ferry can serve that node at any time. So we would require unbounded buffer at that node. So $r_i \leq r_F$ is a necessary condition.

To see why the conditions are sufficient, let $s_i = \frac{r_i}{m}$, then the first condition is equivalent to $\sum s_i < r_F$. One ferry could solve this instance with rates s_i. Consider a stable cyclic solution for this instance with one ferry, with rate r_F. Let this solution take time T for each cycle period (inclusive of waiting times at each node). Start the m ferries in the given solution at points $0, \frac{T}{m}, 2\frac{T}{m}, \ldots, (n-1)\frac{T}{m}$. Pretend that each of these ferries is solving an instance with rates s_i. This is a stable solution, provided that the ferries never run into each other at any node. But notice that $\max s_i = \frac{1}{m} \max r_i \leq \frac{r_F}{m}$. So the ferry spends time at most T/m at each node for the instance with rates s_i. Since the ferries are equally spaced in time, no two ferries would have to serve the same node at a given time. ∎

5 Conclusions

In this paper, we formalize the Message Ferrying model for Mobile ad hoc networks. We characterize stability conditions for problem instances on a node distribution and efficient approximation algorithms for a restricted class of metric node distributions. An interesting question is to extend our results to the more realistic and interesting case of finite ferry rate. Another direction is to generalize the algorithm for a larger class of metrics on which the nodes are distributed. Also, while the ferry problem seems intriguingly similar to the TSP, there seems to be no formal connection. Is there a way to translate the TSP approximation algorithms (on generic metric spaces) to the ferry problem? Also, in this work, we assumed the rate at which a node is producing data to be constant; however this is could be an unreasonable assumption, depending on the application. Is there a natural way to model and solve the more general case?

Acknowledgement. We thank Shiva Kintali for helpful initial discussions.

References

[1] Nikhil Bansal, Avrim Blum, Shuchi Chawla and Adam Meyerson, "Approximation algorithms for deadline-TSP and vehicle routing with time-windows", *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, 166–174, 2004.

[2] Yair Bartal, "Probabilistic approximation of metric spaces and its algorithmic applications", *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*, 184–193, 1996.

[3] Mee Yee Chan and Francis Y. L. Chin, "General Schedulers for the Pinwheel Problem Based on Double-Integer Reduction", *IEEE Transactions on Computers*, Volume 41, Issue 6, 755–768, June 1992.

[4] Mee Yee Chan and Francis Y. L. Chin, "Schedulers for larger class of pinwheel instances", *Algorithmica*, Volume 9, Issue 5, 425–462, May 1993.

[5] Moses Charikar, Chandra Chekuri, Ashish Goel, Sudipto Guha and Serge A. Plotkin, "Approximating a Finite Metric by a Small Number of Tree Metrics", *Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, 379–388, 1998.

[6] Yang Chen, Jeonghwa Yang, Wenrui Zhao, Mostafa Ammar and Ellen Zegura, "Multicasting in Sparse MANETs Using Message Ferrying", *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC) 2006*, Volume 2, 691–696, April 2006.

[7] Thomas D. Dyer and Rajendra V. Boppana, "A Comparison of TCP Performance over Three Routing Protocols for Mobile Ad Hoc Networks", *Proceedings of the 2nd ACM International Symposium on Mobile ad hoc Networking and Computing*, 56–66, 2001.

[8] Jittat Fakcheroenphol, Kunal Talwar and Satish Rao, "A tight bound on approximating arbitrary metrics by tree metrics", *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, 448–455, 2003.

[9] Peter C. Fishburn, P. C. and Jeffrey C. Lagarias, "Pinwheel scheduling: achievable densities", *Algorithmica*, Volume 34, Issue 1, 14–38, 2002.

[10] Robert Holte, Al Mok, Louis Rosier, Igor Tulchinsky and Donald Varvel, "The pinwheel: A real-time scheduling problem", *Proceedings of the 22nd Hawaii International Conference of System Sciences*, 693–702, January 1989.

[11] Robert Holte, Louis Rosier, Igor Tulchinsky and Donald Varvel, "Pinwheel scheduling with two distinct numbers", *Mathematical Foundations of Computer Science 1989* (*Porpolhk abka-Kozubnik 1989*), *Lecture Notes in Computer Science*, Volume 379, 281–290, 1989, and *Theoretical Computer Science* Volume 100, Issue 1, 105–135, 1992.

[12] Shun-Shii Lin and Kwei-Jay Lin, "A pinwheel scheduler for three distinct numbers with a tight schedulability bound", *Algorithmica*, Volume 19, Issue 4, 411–426, 1997.

[13] Viswanath Nagarajan and R. Ravi, "Minimum vehicle routing with a common deadline", *Proceedings of the 9th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX) 2006*, 212–223, 2006.

[14] Christos H. Papadimitriou and Mihalis Yannakakis, "The traveling salesman problem with distances one and two", *Mathematics of Operations Research*, Volume 18, Issue 1, 1–11, February 1993.

[15] Barath Petit, Mostafa Ammar and Richard Fujimoto, "Protocols for Roadside-to-Roadside Data Relaying over Vehicular Networks", *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC) 2006*, Volume 1, 294–299, April 2006.

[16] Jeonghwa Yang, Yang Chen, Mostafa Ammar and Chung-Ki Lee, "Ferry Replacement Protocols in Sparse MANET Message Ferrying Systems", *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC) 2005*, Vol. 4, 2038–2044, March 2005.

[17] Wenrui Zhao, Mostafa Ammar and Ellen Zegura, "A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks", *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, 187–198, May 2004.

[18] Wenrui Zhao, Mostafa Ammar and Ellen Zegura, "Controlling the Mobility of Multiple Data Transport Ferries in a Delay-Tolerant Network", *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM 2005)*, Volume 2, 1407–1418, 2005.