Message Ferrying: Proactive Routing in Highly-partitioned Wireless Ad Hoc Networks

Wenrui Zhao and Mostafa H. Ammar
Networking and Telecommunication Group
College of Computing, Georgia Institute of Technology
Atlanta, Georgia 30332
{wrzhao, ammar}@cc.gatech.edu

Abstract

An ad hoc network allows devices with wireless interfaces to communicate with each other without any pre-installed infrastructure. Due to node mobility, limited radio power, node failure and wide deployment area, ad hoc networks are often vulnerable to network partitioning. A number of examples are in battlefield, disaster recovery and wide area surveillance. Unfortunately, most existing ad hoc routing protocols will fail to deliver messages under these circumstances since no route to the destination exists. In this work, we propose the Message Ferrying or MF scheme that provides efficient data delivery in disconnected ad hoc networks. In the MF scheme, nodes move proactively to send or receive messages. By introducing non-randomness *in a node's proactive movement and exploiting such non-randomness to deliver messages, the MF scheme improves data delivery performance in a disconnected network. In this paper, we propose the basic design of the MF scheme and develop a general framework to classify variations of MF systems. We also study ferry route design problem in stationary node case which is shown to be NP-hard and provide an efficient algorithm to compute ferry route.*

1 Introduction

An ad hoc network allows devices with wireless interfaces to communicate with each other without any pre-installed infrastructure. Many routing algorithms have been developed for ad hoc networks [5, 7, 13, 11, 12, 14, 4]. However, most of these algorithms are designed for fully connected ad hoc networks assuming persistent end-to-end connectivity between any two nodes. While it is desirable to maintain a connected network, ad hoc networks are often vulnerable to network partitioning. Node mobility, limited radio range, physical obstacles, severe weather or other physical factors, might preclude nodes from communicating with others and keep the network in partitioned state. For example, in a battlefield, nodes equipped with short range radio may move out of range of others. In addition, the wide physical range of the deployed area in some circumstances will also prevent full coverage due to the cost involved. Under these environments, most existing routing algorithms will fail to deliver messages to their destinations since no route is found due to network partition. This raises the question of how to deliver data in a constantly disconnected network. While certain applications, like real-time video or audio streaming, require bandwidth or delay guarantees to be useful, other applications would benefit from the eventual and timely delivery of data, even in the presence of network partition. A number of these applications include messaging, file transfer, email and other non real-time applications. We expect that this kind of highly-partitioned ad hoc networks would not be uncommon in practice, e.g. in battlefield, disaster recovery and wide area surveillance. Thus, efficient routing algorithms must be developed to route messages to their destinations despite of network partition.

The existence of network partitioning requires a new approach other than the traditional "store-and-forward" routing paradigm used in most current ad hoc routing algorithms, in which messages are dropped if no route is found to reach the destinations. Instead, a device should buffer and carry the messages until it has a chance to forward them. Thus, devices need to *store*, *carry* and *forward* messages in highly-partitioned ad hoc networks [15, 2].

Several schemes have been proposed to solve the routing problem in highly-partitioned wireless ad hoc networks [15, 2, 8]. Epidemic routing [15] is a flooding-style algorithm in which nodes forward messages to other nodes they meet. This scheme will deliver all messages given unlimited time and memory. The blind-flooding of epidemic routing transmits a lot of redundant messages and requires a large amount of buffering, which leads to poor scalability. The work in [2] proposed an improved scheme over epidemic routing by exploiting node mobility statistics. Nodes estimate the probabilities of meeting other nodes in the future and forward messages based on these probabilities. It can deliver more messages, as compared to epidemic routing, in networks with limited memory. Both schemes in [15] and [2] are *reactive* in that when disconnected, nodes passively wait for their chances to re-connect, e.g. meet with other nodes, which may lead to significant and even unacceptable transmission delays and low throughput. To address this problem, the work in [8] proposed a proactive scheme in which mobile nodes actively modify their trajectories in order to transmit messages as soon as possible. The proposed scheme computes a trajectory for sending messages through intermediate nodes to reach the destinations and optimizes both the transmission delay and the trajectory modification for the case where a single message is transmitted in the network. However, the work in [8] assumes that nodes have full knowledge of the location and movement of all nodes in the network, and cannot support parallel transmission of multiple messages in the network. These assumptions make it difficult for use in realistic networks.

In this work, we propose a new proactive routing scheme for disconnected wireless ad hoc networks. The new scheme is referred to as Message Ferrying, inspired from its real life analog. In this scheme, nodes move proactively in order to send or receive messages. The main idea of the MF scheme is to introduce *non-randomness* in the proactive movement of nodes and exploit such non-randomness to help deliver messages. In the MF scheme, a set of devices called *message ferries* (or ferries in short) take responsibility for carrying messages between disconnected nodes. Message ferries move around the deployed area according to known routes and communicate with other nodes they meet. With knowledge of the ferry routes, nodes can adapt their trajectories to meet the ferries and transmit or receive messages. By using ferries as relays, nodes can communicate asynchronously with other nodes that are disconnected. The MF scheme differs significantly from previous schemes in its *non-random* proactive movement of nodes. With proactive movement of both the ferries and the nodes, the MF scheme provides regular connectivity in an otherwise disconnected ad hoc network.

In this paper, we propose the basic design of the MF scheme and develop a general framework to classify variations of MF systems. We then study the ferry route design problem for the case where nodes are stationary. Since the MF scheme depends on node movement to deliver messages, the choice of ferry routes will have significant impact on the achieved throughput and delay. We formulate this problem as an optimization problem and divide it into two subproblems, one of which is shown to be NP-hard. By attacking each of these subproblems we develop an efficient algorithm to compute ferry routes.

The rest of this paper is organized as follows. We introduce the Message Ferrying scheme in Section 2. In Section 3 we describe the ferry route design problem in the stationary node case and present an algorithm to compute ferry routes. Section 4 presents our preliminary simulation results. We conclude the paper in Section 5.

2 Message Ferrying Scheme

The Message Ferrying (MF) scheme is a proactive approach for routing in disconnected ad hoc networks. It addresses the disconnection problem by introducing *non-randomness* to node mobility and exploiting such non-randomness to provide connectivity. In the MF scheme, the network devices are classified as message ferries (*ferries* for short) or regular nodes based on their roles in communication. Ferries are devices which take responsibility of carrying messages between disconnected nodes, while regular nodes are devices without such responsibility. Ferries move around the deployed area according to known routes, collect messages from the sending nodes and deliver messages to their destinations or other ferries. With knowledge about ferry routes, nodes can adapt their trajectories to meet the ferries and transmit or receive messages. Ferries actually serve as "rendezvous points" for message senders and receivers. By using ferries as relays, nodes can communicate with others that are disconnected.

Figure 1 shows an example of how message transfer is done using a single ferry in a mobile ad hoc network. The ferry moves on a known route which is depicted using dashed line in Figure 1. In Figure 1(A), the sending node *S* actively approaches the ferry and forwards its messages to the ferry which will be responsible for delivery. In Figure 1(B), the receiving node *R* communicates with the ferry and receives its messages.

2.1 Functions and Capabilities of MF Scheme Components

As discussed above, message ferrying is a broad concept, with the potential for many variations in specific design and implementation. We develop a general framework for MF systems based on the capabilities and functions performed by the various components of a message ferrying system. We consider the following five dimensions that define the context for a Message Ferrying scheme:

1. *Ferry Mobility:* The movement of the ferry is a critical feature of an MF scheme. We consider two possible cases, which may co-exist in the same scenario:

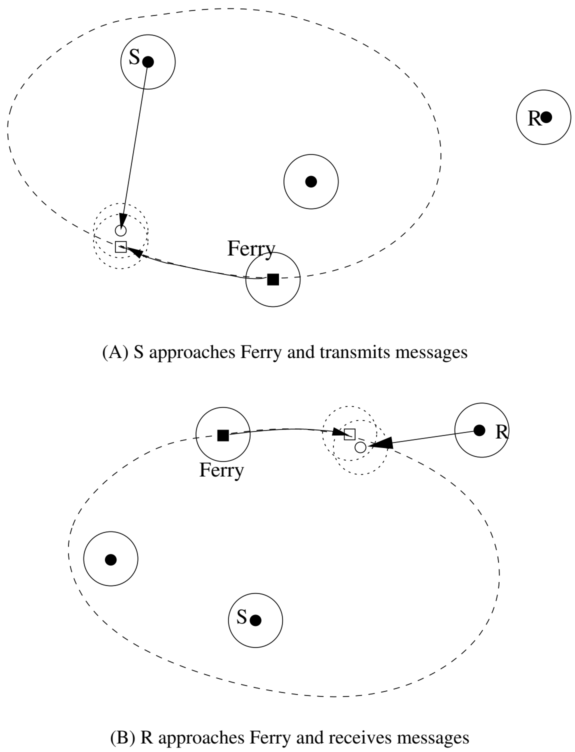

Figure 1. An example of message delivery using a ferry as relay.

- The ferry mobility is determined for non-messaging reasons: For example, when piggybacking a ferry on a metropolitan area bus, the route the ferry takes is determined based on passenger-carrying concerns and not message-forwarding concerns.
- The ferry mobility is specifically designed for improving the performance of messaging. For example, the ferry is implemented in a subset of robots dispersed in a disaster area, and the mobility of the ferry robots is specifically optimized for maximizing the efficiency of messaging among the other robots.

2. *Regular Node Mobility:* While the ferry is always a mobile entity, the regular nodes can be stationary or mobile. Similar to ferry mobility, mobile regular nodes can move for non-messaging reasons or specifically to improve messaging performance.

3. *Number of Ferries and Level of Coordination:* In general, an MF system may have multiple ferries, each with a possibly different set of capabilities. Ferries may operate completely independently of each other or their movements may be coordinated. In this latter case, a message might be forwarded through multiple ferries while being routed to its destination.

4. *Level of Regular Node Coordination:* The regular nodes can use the ferry to deliver data in two fundamentally different ways:
   - The regular nodes operate independently, with each regular node acting on its own and in charge of delivering to and receiving from the ferry.
   - The regular nodes coordinate with each other to form connected *clusters*. Within a cluster, one or more *gateway nodes* are in charge of communicating with the ferry. The other nodes communicate with these gateway nodes using traditional ad-hoc network routing protocols.

5. *Ferry Designation:* Ferries can be either specially designated nodes or regular nodes that are (perhaps temporarily) elevated to perform ferry functions. In the former case, it may be possible to assume that a ferry's resources (power, memory, disk storage) are not as limited as typical nodes. For the latter case, there is, of course, the question of when and how to change node designation.

In this paper, we focus on the case where regular nodes are stationary and a single ferry is used. The study of other variations of message ferrying, such as in mobile networks and the use of multiple ferries, is deferred to future research.

3 Ferry Route Design for Stationary Node Case

We describe the basic design of the MF scheme in the previous section. In this section, we focus on the ferry route design problem in the stationary node case. In this case, nodes are located at fixed positions and network partition prevents some nodes from communicating with others. The ferries are used to relay messages between disconnected nodes. A basic question is how to design good or optimal ferry routes, given node positions and their communication requirements. In this work, we assume that a *single* ferry is deployed which moves at a constant speed and the expected traffic between nodes is known in advance.

Let $N = n_1, n_2, \ldots, n_k$ be a set of k nodes that want to communicate. The ferry moves at a constant speed f. Let d be the transmission range of the nodes and the ferry[1]. The nodes and the ferry communicate via a wireless channel when they are within range of each other. Assuming

[1] The results and algorithm developed in this section can be easily extended to handle the case where the nodes and the ferry have different transmission ranges.

that communication between the nodes and the ferry uses different radio frequencies in each direction, sending and receiving can take place simultaneously. Let r be the transmission rate between the nodes and the ferry. When the ferry is in range of multiple nodes, some policy is used to schedule the transmission and reception of nodes. Instead of focusing on specific MAC scheduling policies, we define a scheduling policy in a more general sense. Let Γ be the power set of N. Then a policy P can be defined as a function $P : N \times \Gamma \rightarrow [0, 1]$. For a set of nodes A and a node n_i, $P(n_i, A)$ defines the portion of transmission time allocated to node n_i when the ferry is within range of nodes in A. Obviously, we have $P(n_i, A) = 0$ if $n_i \notin A$ and $\sum_{i=1}^{k} P(n_i, A) = 1$.

Let b_{ij} be the expected traffic from node n_i to n_j, measured in bits per second. Define $s_i = \sum_{j=1}^{k} b_{ij}$ and $r_i = \sum_{j=1}^{k} b_{ji}$, which are the total incoming and outgoing traffic for n_i respectively. For any ferry route T, we denote its length as $|T|$. Let d_{ij}^T be the average delay for traffic from node n_i to n_j in route T. Then the average delay for all traffic can be defined as

$$d^T = \frac{\sum_{1 \le i,j \le k} b_{ij} d_{ij}^T}{\sum_{1 \le i,j \le k} b_{ij}}$$

.

The ferry route problem consists of finding an optimal route T such that the bandwidth requirements b_{ij} are met and the average delay d^T is minimized. Rather than addressing the combined problem, we break it into two sub-problems. The first one seeks to find a route that minimizes the average delay for the expected traffic matrix without considering the bandwidth requirements. The second sub-problem extends the route generated in the first sub-problem, if necessary, to meet the bandwidth requirements.

3.1 Delay problem

In the first sub-problem, we only consider the message delay. The average delay d_{ij}^T for messages from node n_i to node n_j consists of the waiting time w_{ij} at n_i before transmitted to the ferry, and the carrying time c_{ij} at the ferry before delivered to n_j. With the assumption of constant bit rate traffic, w_{ij} equals to $\frac{|T|}{2f}$. For simplicity, we measure the carrying time c_{ij} as the time for the ferry to move from n_i's position to n_j's position. Let l_{ij}^T be the distance from nodes n_i to n_j in route T. Then $d_{ij}^T = \frac{|T|}{2f} + \frac{l_{ij}T}{f}$. The delay sub-problem, denoted as MFR-delay problem, can be defined as follows.

Definition (MFR-delay problem) Let N be a set of nodes and b_{ij} be the expected traffic from node n_i to node n_j, the MFR-delay problem consists of finding an optimal route T such that T visits all nodes in N and the average delay d^T is minimized.

Property 1. If $b_{ij} = b_{ji}$ for all i and j, then $d^T = \frac{|T|}{f}$ for any route T.

2. If route T' is the reverse route of T, that is in T' the ferry visits the nodes in the reverse order as in T, then $d^T + d^{T'} = \frac{2|T|}{f}$. Thus either d^T or $d^{T'}$ is no more than $\frac{|T|}{f}$.

3. Let T be a shortest route that visits all nodes in N, e.g. an optimal traveling salesman tour for N. If O is an optimal route for any MFR-delay problem defined on nodes N, then $\frac{|T|}{2f} \le d^O \le \frac{|T|}{f}$.

Proof:

1. $d_{ij}^T + d_{ji}^T = \frac{|T|}{f} + \frac{l_{ij}^T}{f} + \frac{l_{ji}^T}{f} = \frac{2|T|}{f}$. So, by the definition of d^T, it is easy to prove that $d^T = \frac{|T|}{f}$.

2. We have $d_{ij}^T + d_{ij}^{T'} = \frac{|T|}{2f} + \frac{l_{ij}^T}{f} + \frac{|T'|}{2f} + \frac{l_{ij}^{T'}}{f} = \frac{2|T|}{f}$ since $|T| = |T'|$ and $l_{ij}^T + l_{ij}^{T'} = |T|$. Thus $d^T + d^{T'} = \frac{2|T|}{f}$. Both d^T and $d^{T'}$ are positive, so either d^T or $d^{T'}$ is no more than $\frac{|T|}{f}$.

3. Let T' be the reverse route of T. By Property 2, either d^T or $d^{T'}$ is not more than $|T|/f$. Since O is optimal, $d^O \le \frac{|T|}{f}$. $d_{ij}^O = \frac{|O|}{2f} + \frac{l_{ij}^O}{f} \ge \frac{|T|}{2f} + \frac{l_{ij}^T}{f} \ge \frac{|T|}{2f}$. Thus $d^O \ge \frac{|T|}{2f}$. Therefore, we have $\frac{|T|}{2f} \le d^O \le \frac{|T|}{f}$.

■

Theorem 1 *The MFR-delay problem is NP-hard.*

Proof: Property 1 shows that when the traffic matrix is symmetric, the optimal route for the MFR-delay problem has shortest length. So we can reduce the Euclidean Traveling Salesman Problem to the MFR-delay problem. Since the Euclidean Traveling Salesman Problem is NP-complete [10], we prove that the MFR-delay problem is NP-hard. ■

The above Property and Theorem 1 relate the MFR-delay problem to the traveling salesman roblem (TSP) and lead us to adapt solutions from the well-studied TSP. We adopt a two-phase approach to solve the MFR-delay problem. First a starting route is generated using some TSP algorithm, e.g. nearest neighbor, greedy, or LKH [1, 6]. Since TSP algorithms only try to optimize the route length, instead of the message delay, we can further reduce the average delay of the route by applying delay-based local optimization techniques similar to the 2-opt and 2H-opt techniques [1] in TSP. These delay-based techniques, denoted as 2-opt-delay and 2H-opt-delay respectively, work as follows.

- *2-opt-delay.* A 2-opt swap removes two edges AB and CD from the route and replaces them with edges AC and BD while maintaining the route. This technique tries to reduce the delay of the route using 2-opt swaps until no better route can be found.

- *2H-opt-delay.* A 2H-opt swap moves a node from one position in the sequence of route to another. This technique tries to reduce the delay of the route using 2-opt and 2H-opt swaps until no better route can be found.

The relation between route length and route delay shown in the above Property justifies our use of TSP algorithms in our approach. The following theorem provides an upper bound for the delay obtained by this approach.

Theorem 2 *Let O is an optimal route for the MFR-delay problem. Let T be the route generated by a TSP algorithm with approximation ratio α. Then the resulting delay of route T satisfies $d^T \leq 2\alpha d^O$.*

Proof: Let T' be an optimal TSP tour. Then $d^O \geq \frac{|T'|}{2}$ by Property 3. Thus, from Property 2, we have $d^T \leq |T| \leq \alpha|T'| \leq 2\alpha d^O$. ■

3.2 Bandwidth problem

Given the route T^* generated in the first sub-problem, we now consider how to extend T^* to meet a node's bandwidth requirement. For any route, the achieved bandwidth of a node is αr where α is the fraction of the ferry route in which the node is in transmission and r is the transmission rate between the ferry and the node. Thus to increase a node's bandwidth we need to increase its transmission time which is also affected by the scheduling policy used. Since increasing a node's transmission time leads to longer ferry route and larger delay for other traffic, we should minimize the amount of route extension.

We formulate the bandwidth sub-problem as a linear programming (LP) problem as follows. We first decompose T^* into segments by cutting T^* when it enters or leaves a node's coverage area and index these segments as $1, 2, \ldots, m$. Let v_i^s be the allocated transmission time for node n_i in route T^*. Define a $k \times m$ matrix F^s such that the entry f_{ij}^s of F^s is the portion of transmission time allocated to node n_i when the ferry is moving on segment j according to the scheduling policy. Similarly, we define the variable v_i^r and the matrix F^r for the receiving direction[2]. Let x_i, $1 \leq i \leq m$, be the extra time the ferry spends on segment i for bandwidth extension. To meet node n_i's bandwidth requirement for transmission, we have

$$\frac{(\sum_{j=1}^{m} f_{ij}^s x_j + v_i^s) r}{\frac{|T^*|}{f} + \sum_{j=1}^{m} x_j} \geq s_i$$

Similarly we can derive constraints on the receiving direction. After transformation, we get the following linear programming problem which can be solved efficiently using methods like Simplex [9].

$$\begin{aligned} \text{minimize} \quad & \sum_{j=1}^{m} x_j, \qquad (1) \\ \text{subject to} \quad & \sum_{j=1}^{m} (f_{ij}^s - \frac{s_i}{r}) x_j \geq \frac{s_i |T^*|}{fr} - v_i^s, \\ & \sum_{j=1}^{m} (f_{ij}^r - \frac{r_i}{r}) x_j \geq \frac{r_i |T^*|}{fr} - v_i^r, \\ & 0 \leq x_j \text{ and } 1 \leq i \leq k. \end{aligned}$$

By combining the solutions for the two sub-problems, we now present the algorithm, called MFR algorithm, for computing ferry routes which is shown in Figure 2. The performance of MFR Algorithm in disjointed node case can be established as follows.

MFR Algorithm

1. Compute a TSP tour T^* for nodes in N using a TSP approximation algorithm.

2. Apply 2H-opt-delay local optimization technique to T^* to reduce the average delay.

3. Generate the linear programming problem as in Formula 1 for route T^* and solve it. Let $X = \{x_1, x_2, \ldots, x_m\}$ be the optimal solution.

4. Modify T^* to meet each node's bandwidth requirement by taking a detour of length $x_i f$ on segment i.

Figure 2. Algorithm for computing ferry route.

Theorem 3 *Let T be the route generated by the MFR Algorithm and α be the approximation ratio of the TSP algorithm used in the MFR Algorithm. Route O is an optimal route for the ferry route problem and f is the speed of the ferry. Let d be the transmission range of the nodes and the ferry. If the distance between any two nodes is at least $2d$, e.g. the coverage areas of nodes are disjointed, then*

$$d^T \leq 2((1 + \frac{8}{\pi})\alpha + 1)d^O + \frac{8d\alpha}{f}$$

.

[2] Note that different scheduling policies can be used for message transmission and reception.

Proof: Let T' be the tour generated by the TSP algorithm in the MFR Algorithm and O' be an optimal TSP tour. Then we have $|T'| \le \alpha|O'|$. From the description of the MFR Algorithm, we have $|T| = |T'| + e \le \alpha|O'| + |O|$ where e is the amount of tour extension to meet all nodes' bandwidth requirements.

We now prove our claim using the technique proposed in [3]. Let A be the area swept by a disk of radius $2d$ whose center moves on route O. Since route O visits each node's coverage area, A includes the coverage area of each node. We have $k\pi d^2 \le A \le 4d|O| + 4\pi d^2$ where k is the number of nodes. Thus, $k \le \frac{4|O|}{\pi d} + 4$. Note that O may not visit each node's position since the ferry can transmit to nodes within distance d. A TSP tour which visits each node's position can be obtained from route O by making a detour of length of at most $2d$ to visit each node's position. This tour is of length $|O| + 2kd \le (1 + \frac{8}{\pi})|O| + 8d$. Since O' is an optimal TSP tour, $|O'| \le (1 + \frac{8}{\pi})|O| + 8d$. Therefore, we have $|T| \le \alpha|O'| + |O| \le ((1 + \frac{8}{\pi})\alpha + 1)|O| + 8d\alpha$.

By Property 2, $d^T \le \frac{|T|}{f}$. For route O, since the average waiting time of messages in the sending node is $\frac{|O|}{2f}$, $d^O \ge \frac{|O|}{2f}$. So we prove $d^T \le 2((1 + \frac{8}{\pi})\alpha + 1)d^O + \frac{8d\alpha}{f}$. ∎

When the number of nodes is large, $\frac{8d\alpha}{fd^T}$ will be close to 0. Thus the MFR Algorithm has an approximation ratio of $2((1 + \frac{8}{\pi})\alpha + 1)$ for large networks with disjointed nodes.

4 Preliminary Experimental Results

In this section we present preliminary experimental results on the performance of the MF scheme in the stationary node case. We model n nodes distributed in a rectangular area $4000m \times 4000m$ in dimension. A single ferry is deployed which moves at a speed of $20m/s$. The transmission range of both the nodes and the ferry is $100m$ unless otherwise noted and the transmission rate is 10Mbps. We simulate a fair sharing scheduling policy in which each node gets equal access to the ferry when the ferry is within range of multiple nodes. Two kinds of node distributions are considered in the experiments.

- *Random Uniform Node Distribution (UN).* In this distribution, each node's x and y coordinates are chosen randomly from interval $(0, 4000)$.

- *Random Clustered Node Distribution (CN).* In this case, we first compute $n/20$ cluster centers whose coordinates are chosen randomly from interval $(0, 4000)$. For each of the n nodes, we calculate its position by computing two normally distributed variables, each multiplied by $4000/\sqrt{n}$, and adding them to the coordinates of a randomly chosed cluster center.

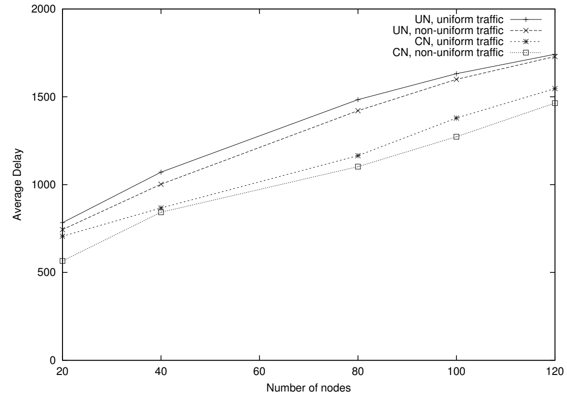

Figure 3. Message delay vs. network size

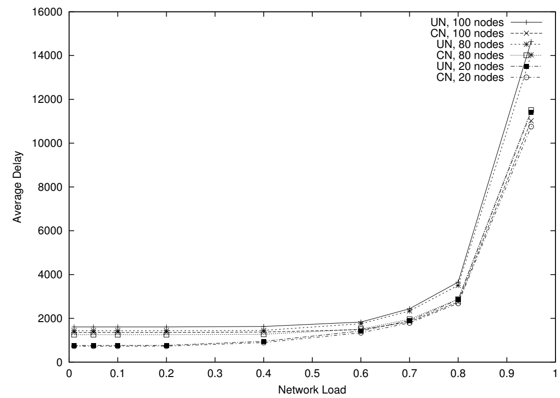

Figure 4. Message delay vs. network load

We also experiment with different traffic models in the simulations. In the uniform traffic model, the traffic between any two nodes has equal bandwidth which is set to $\frac{10}{n-1}$Kbps. In the non-uniform traffic model, $0.1n$ flows are sending at rate $\frac{5}{n-1}$Mbps and the rest are sending at $\frac{10}{n-1}$Kbps. This non-uniform traffic model represents the case where some flows transmit much more data than others. In both traffic models flows are generated at constant bit rates.

Given node locations and the traffic models, we use the MFR Algorithm to compute the ferry route. In our implementation of the MFR Algorithm, we choose the nearest neighbor algorithm with 2H-opt to compute the starting TSP tour and then use local optimization technique 2H-opt-delay to reduce the message delay.

We evaluate the performance of the MF scheme under different network sizes and traffic loads. The main metric used in our evaluations is average message delay. Figure 3 shows the average message delay with different network sizes. In general, as the number of nodes increases, so does the message delay. And the clustered node distributions tend to have smaller delay as compared to the uniform node distributions. This is because clustering of nodes tends to shorten the ferry route which results in smaller message delay. The important point is that, in the scenarios we study here, *the message ferry routing can achieve reasonable performance even when the nodes are extremely disconnected*. For example, with 40 nodes randomly distributed in a $4000m \times 4000m$ area, each node can send messages at rate $10Kbps$ with about 1071 second delay. Without any communication infrastructure, we believe this delay is inherent given the dimension of the deployment area and the limited transmission range.

Figure 4 shows the effect of network load on the message delay. We define network load as the ratio between total transmission bandwidth generated by nodes and the transmission rate of the wireless interface. The average message delay increases as the network load increases especially when the network load is close to 1. However, for modest network load $(40\% - 50\%)$, the increase of delay is very slow suggesting that nodes can send at higher rates without significant delay penalty. We also evaluate the performance of the MF scheme under different transmission ranges which show similar results.

5 Conclusion

In this paper we presented the Message Ferrying scheme to solve the data delivery problem in highly-partitioned wireless ad hoc networks. It is a proactive scheme in which a set of nodes called message ferries take the responsibility of carrying messages and providing connectivity for all nodes in the network. By introducing non-randomness in nodes' proactive movement and exploiting such non-randomness to deliver messages, the MF scheme provides regular connectivity in a disconnected network and improves data delivery performance without global knowledge of each node's location. In this paper, we also study the ferry route design problem for the case where nodes are stationary. We show that this problem is NP-hard and provide an efficient algorithm to compute the ferry route.

References

[1] J. Bentley. Fast algorithms for geometric traveling salesman problems. *ORSA Journal on Computing*, 4:387–411, 1992.

[2] J. Davis, A. Fagg, and B. Levine. Wearable computers as packet transport mechanisms in highly-partitioned ad-hoc networks. In *International Symposium on Wearable Computing*, October 2001.

[3] A. Dumitrescu and J. Mitchell. Approximation algorithms for tsp with neighborhoods in the plane. In *Twelfth ACM-SIAM Symposium on Discrete Algorithms*, pages 38–47, January 2001.

[4] Z. Haas and M. Pearlman. The performance of query control schemes for the zone routing protocol. In *SIGCOMM*, 1998.

[5] D. Johnson and D. Maltz. Dynamic source routing in ad-hoc wireless networks. In *SIGCOMM*, August 1996.

[6] D. Johnson and L. McGeoch. Experimental analysis of heuristics for the stsp. *The Traveling Salesman Problem and its Variations*, 2002.

[7] Y. Ko and N. Vaidya. Location-aided routing (lar) in mobile ad hoc networks. In *MOBICOM*, November 1998.

[8] Q. Li and D. Rus. Sending messages to mobile users in disconnected ad-hoc wireless networks. In *MOBICOM*, August 2000.

[9] G. Nemhauser, A. R. Kan, and M. T. (editors). Optimization. 1989.

[10] C. Papadimitriou. The euclidean traveling salesman problem is np-complete. *Theoret. Comput. Sci.*, 4:237–244, 1977.

[11] V. Park and M. Corson. A highly adaptive distributed algorithm for mobile wireless networks. In *INFOCOM*, April 1997.

[12] C. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. *Computer Communication Review*, 24, October 1994.

[13] C. Perkins and E. Royer. Ad hoc on-demand distance vector routing. In *2nd IEEE Workshop on Mobile Computing Systems and Applications*, February 1999.

[14] R. Sivakumar, P. Sinha, and V. Bharghavan. Cedar: a core-extraction distributed ad hoc routing algorithm. *IEEE Journal on Selected Areas in Communications*, 17:1454–1465, August 1999.

[15] A. Vahdat and D. Becker. Epidemic routing for partially-connected ad hoc networks. *Technical report, Duke University*, 2000.