Scheduling Mobile Nodes for Cooperative Data Transport in Sensor Networks *

Reza Moazzez-Estanjini,† Jing Wang,‡ and Ioannis Ch. Paschalidis§

***Abstract*—Message Ferrying has been shown to be an effective approach to support routing in sparse ad hoc or sensor networks. Considering a generic network model where each node in the network wishes to send data to some (or possibly all) other nodes with known (and possibly different) rates, we propose three schemes enabling multiple ferries to coordinate in collecting and delivering the data. We analyze the performance of each scheme and establish bounds on the average and worst-case delay. The latter bounds are useful in offering performance guarantees. We establish that under one of our schemes, constant per-node throughput is achievable within constant maximum (worst-case) delay as the network size grows. Using simulation, we compare our proposed schemes with an alternative, the Ferry Relaying algorithm proposed earlier in the literature. The results show that our schemes perform better and provide guidance on which scheme to use given performance preferences and the number of available ferries.**

***Index Terms*—Wireless sensor networks, delay tolerant networks, sparse ad hoc networks, message ferrying, scheduling.**

I. Introduction

In traditional ad-hoc or sensor networks, communication between two nodes requires the discovery of a connected path utilizing other nodes as relays. In some situations, due to a damaged infrastructure (e.g., in disaster relief situations) or due to limited wireless range (e.g., sparse sensor networks deployed in remote areas), connectivity between nodes is intermittent at best, and connected paths between pairs of nodes may not consistently exist. As a result, traditional ad hoc and mesh networking routing protocols are not applicable. In addition, relaying data over a large number of hops significantly reduces the lifetime of sensor nodes.

An alternative way to establish connectivity and transport data is to use *mobile nodes* for which we will adopt the term *Message Ferries (MFs)*. MFs can be robots, Unmanned Aerial Vehicles (UAVs), Autonomous Underwater Vehicles (AUVs), or any type of vehicle which is tasked with visiting the stationary nodes (or *nodes*, henceforth) and physically transporting data from node to node as needed. Specifically, the MFs have onboard memory and can act as relays, reaching to the nodes, downloading the data, mechanically carrying them to the destination, and uploading them when in close proximity to the intended sink. The use of MFs allows communications to be short-range (between nodes and MFs) at a drastically reduced energy budget. Furthermore, MFs can easily be recharged periodically at a base station as opposed to nodes which may not possess energy replenishment capabilities. The trade-off to these benefits is increased delay as physical data transport by the MFs can be quite slow. However, for many *delay-tolerant* applications this is acceptable.

A variety of applications have been considered for sensor networks served by MFs as we described. For instance, a sparse sensor network can be deployed in a remote inhospitable area with no communication infrastructure (wireline or wireless). Some examples include sensors in border surveillance missions, sensors for environmental observation [1], sensors underwater [2], or sensors tracking wildlife [3]. In a different scenario, a myriad of sensors can be deployed in an urban area for monitoring air/water quality, traffic, trash cans [4], or the condition of traffic meters. Similarly, sensors can be used for infrastructure/machinery condition monitoring. In many of these settings the embedded sensors can not afford longer-range wireless communications because either they are too sparse, inexpensive, energy limited, or have significant bandwidth needs (e.g., cameras). The use of MFs for data transport offers a viable solution. This is facilitated by the fact that in some of the environments we considered (e.g., urban) vehicles already exist (e.g., buses, municipal vehicles, taxis, police cars) and can easily be tasked to act as MFs.

Since the data is transported via the MFs, the design of the ferries' routes has a significant impact on network performance. Therefore, a basic question we address is how to design effective routes for the ferries, given node positions and their communication requirements. We will consider schemes that utilize multiple MFs so that one can scale the size of the network to be served. An MF can possibly use other MFs as relays to deliver a message. This requires some synchronization; that is, MFs are required to be physically close to each other at certain times in order to communicate. Alternatively, they can use special nodes to drop and pickup messages. In either case, there is some cost: the synchronization overhead in the former case and the hardware cost of the special nodes in the latter. We elect to focus on the former case even though several of our results apply to the latter as well.

Our main contribution is the development of three alternative schemes for ferry coordination and route design. For each of the proposed schemes we present bounds on the average and worst-case delay. The latter worst-case analysis

* Research partially supported by the NSF under grant EFRI-0735974, by the DOE under grant DE-FG52-06NA27490, by the ARO under grant W911NF-11-1-0227, and by the ODDR&E MURI10 program under grant N00014-10-1-0952.

† Center for Information & Systems Eng., Boston University, e-mail: reza2007@bu.edu

† Center for Information & Systems Eng., Boston University, e-mail: wangjing@bu.edu

§ Dept. of Electrical and Computer Eng., and Div. of Systems Eng., Boston University, 8 St. Mary's St., Boston, MA 02215, yannisp@bu.edu, http://ionia.bu.edu/.

was largely missing from the related literature, which we review in Section II. Clearly, being able to compute worst-case delay bounds allows us to offer performance guarantees, which is extremely useful in network design. We also treat some limited cases where "bounded" stochasticity is present in the problem data (communication rates, travel times). Using simulation, we highlight advantages and disadvantages of each scheme under different scenarios while, at the same time, compare each of our schemes with an alternative from the literature. Another contribution of our work is to show that even if each node sends data to multiple other nodes, Message Ferrying is still *scalable* in the sense that it achieves constant per-node throughput within constant delay.

In what follows, Section II reviews related work. Section III states the problem and the assumptions. Sections IV, V, and VI introduce the three proposed schemes and present the corresponding results. Section VII includes our simulation results, and conclusions are drawn in Section VIII.

II. Related Work

Several works use mobile nodes to carry data in networks, each adopting a different term to refer to them: *data MULE* [5], *Message Ferry* [6], [7], *Actor* [8], *Actuator* [9], sensor node [10], or wireless agent [11].

In terms of applications, a number of recent projects promote the effectiveness of using message ferries, e.g., the ZebraNet project [3], the Manatee project [12], the DakNet project [13], unmanned vehicles in underwater environmental monitoring [2], and a UAV (Unmanned Aerial Vehicle) in structural health monitoring [14]. A host of other applications are reviewed in [6], [7], [5], [15], [16].

The topic of ferry route design, which is the focus of our work, was first thoroughly investigated in [6] where the authors considered a single MF. [17] considered multicasting using a single ferry moving on a pre-determined route. [15] considered multiple MFs in a stationary network scenario and proposed a so called *Ferry Relaying Algorithm (FRA)* which allows MFs to exchange data between each other. FRA requires synchronization between MFs since they have to be physically close to each other at certain times in order to communicate. The schemes we propose in this paper fall in the same category as data gets relayed among the MFs. As we show, our schemes outperform FRA in a number of scenarios and offer advantages such as guaranteeing worst-case delay and being scalable.

We also note that the problem we consider is fundamentally different from any variant of the *Pickup and Delivery Problem* (PDP) [18], [19], [20]. In PDP, the quantities of goods (here, the data) are considered to be discrete and are waiting in their origin locations to be picked up and delivered to a depot. Once the tour is completed and the vehicle returns to the depot where it started its trip, the problem is over. In our setting, on the other hand, the goods (the data) at each origin location (nodes) are constantly being generated according to some rates and the objective is to minimize the *average* or *worst-case* data delivery delay. The data are buffered in each node until the location is visited by an MF, hence, the amount of data to be carried depends on the time that the node is visited by the MF. Further, in PDP, delays typically consist of just travel times, whereas in our problem loading and unloading (i.e., transmission) delays are not ignored.

III. Problem Statement and Assumptions

We consider a network with N (stationary) nodes. We assume that each node generates data at a constant rate destined for other nodes. This corresponds to a situation where the sensor network is monitoring/actuating some underlying physical system whose dynamics can be captured by a (sufficiently high) constant rate. Let $\lambda(i,j)$ be the rate at which node i generates data to be sent to node j, for all j. Data have a single destination and we do not consider multicast scenarios. We will assume that data in the network are exclusively transported by the MFs. In practice, one would typically see disconnected clusters of sensors communicating via the MFs. Representing each cluster by a single node yields a network using only MFs for data transport.

For most of our results we will assume that the $\lambda(i,j)$'s are known (estimated) with certainty. We will also discuss some cases where $\lambda(i,j)$ is an upper bound with probability one (w.p.1) to the stochastic and potentially time-varying rate from i to j. We have M MFs at our disposal. For simplicity, the communication between MFs and nodes is assumed to be done at zero distance and at a rate of W_1 *bps* while communication between any two MFs is done at a rate of W_2 *bps* ($W_2 \geq W_1$). Although in many situations the transmission of data between MFs and nodes might be faster than the MF traveling time, similar to [15] we take the more general view that MFs have limited bandwidth and accounting for transmission times is important for carrying out an accurate analysis. We assume that the MFs move at a constant speed of V *m/s*. The data are buffered at each node until an MF comes and collects them. The objective is to design the MFs' routes such that the delay in delivering the data is minimized. We will only consider periodic ferry routes that form simple cycles between a set of nodes. Specifically, each MF will be assigned to a set of nodes and will cycle periodically through these nodes and potentially some additional points where it meets with other MFs.

IV. Centralized Ferry Relaying: CFR

In this section, we propose the CFR scheme. Fig. 1(a) depicts a simple geographic partitioning of the entire network. There is a single MF assigned to each segment. Within each *cycle*, each MF collects the buffered data from its nodes. Later in the same cycle, all MFs meet at some pre-determined *contact point* and relay their buffered data. All MFs stay at the contact point until all inter-MF data transfers are completed. Note that each MF's route consists of a *Hamiltonian cycle* [1] that includes all its assigned nodes and the contact point; we will refer to this route as the *MF* i *cycle* for each i. The partitioning presented in Fig. 1(a) is a "contiguous" partitioning in the sense that each segment assigned to an MF is geographically contiguous. This is similar to FRA partitioning [15] (in which segments are rectangles with equal

[1] A Hamiltonian cycle is a cycle in an undirected graph which visits each node exactly once and returns to the starting node.

area). However, in the CFR scheme we allow the segments to be "non-contiguous." This allows assigning any combination of nodes to any MF and results in greater flexibility (see Fig. 1(b)). For instance, consider a situation where nodes located farther from each other have higher data rates among themselves, and nodes closer to each other have lower data rates among themselves. Then, we might prefer to allocate the nodes farther from each other to one common MF, trading-off shorter length routes for a decrease in relay time among MFs.

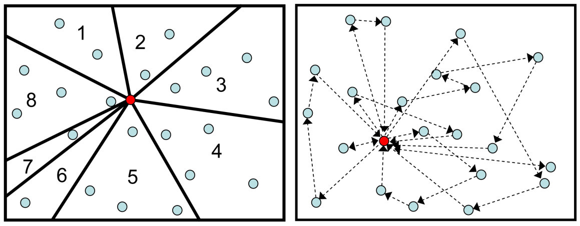


Fig. 1. Examples of partitioning into 8 segments (the blue circles represent the nodes, and the red circle in the middle of the graph represents the contact point); (Left): "Contiguous" partitioning. (Right): CFR scheme ("Non-contiguous" partitioning).

A. Average and worst-case delay

In the CFR scheme, data are relayed between MFs at most once, i.e., if the destination node is in a different MF cycle than that of the source node, then the MF serving the source node collects the data and relays it to the MF serving the destination node. We suggest placing the contact point at the *centroid* (geometric center) of the nodes. Each MF is tasked with: collecting data from its nodes, relaying to other MFs the data destined to their nodes, and delivering to its own nodes the data destined to them. We refer to the time needed by MF i to complete these activities as *MF i's cycle time* and denote it by T_i. (On a notational remark, throughout the paper, the superscript *d* denotes downloading data to the MF under consideration, the superscript *u* denotes uploading data to another MF or node, and the superscript *l* denotes local traffic.) T_i consists of the following terms:

- *Inter-MF relay time* (Y): this is the same for all MFs and is the time needed for all data to be relayed among MFs at the contact point. Note that all the MFs wait in the contact point until all relays among all MFs are finished.
- *MF-node transfer time* (X_i): the transfer time (X_i^d) that MF i needs in order to collect all the data from its nodes, plus the transfer time (X_i^u) that MF i needs in order to deliver all its buffered data that are destined to MF i's nodes. We have $X_i = X_i^d + X_i^u$.
- *MF travel time* (A_i): the time needed to traverse its cycle.

Since MFs exchange data directly, synchronization is required so that MFs meet at the same time at the contact point. For synchronization, the cycle time of the system, T, can be chosen as the maximum of the MFs' cycle times, hence $T = \max_i\{T_i\}$. An alternative to synchronization is to place nodes with ample buffer space at the contact point so that each MF can deposit and upload from this buffer without having to wait for the other MFs. However, due to the cost/technical constraints that these additional nodes would introduce, we will not pursue this direction in this paper. We denote the set of assigned nodes to MF i by $\mathscr{N}(i)$. To calculate the cycle time for a given configuration of the scheme, we have

$$T_i = Y + X_i + A_i, \tag{1}$$

where

$$Y = \frac{T}{W_2}\sum_{k=1}^{M}\sum_{k'=1,k'\neq k}^{M}\sum_{l\in\mathscr{N}(k)}\sum_{l'\in\mathscr{N}(k')}\lambda(l,l'), \tag{2}$$

$$X_i^d = \frac{T}{W_1}\sum_{k=1}^{M}\sum_{l\in\mathscr{N}(i)}\sum_{l'\in\mathscr{N}(k)}\lambda(l,l'), \tag{3}$$

$$X_i^u = \frac{T}{W_1}\sum_{k=1}^{M}\sum_{l\in\mathscr{N}(k)}\sum_{l'\in\mathscr{N}(i)}\lambda(l,l'), \tag{4}$$

$$A_i = L_i/V. \tag{5}$$

L_i in above equation is the length of the MF i cycle. Solving for T we obtain

$$T = \max_i\left\{\frac{A_i}{1-\frac{Y}{T}-\frac{X_i}{T}}\right\}. \tag{6}$$

Note that $\frac{Y}{T}$ and $\frac{X_i}{T}$ do not depend on T.

Let now $\mathscr{S}$ denote the set of all *Source-Destination (SD) pairs* in the network. For each $s\in\mathscr{S}$, D_s denotes the time needed to deliver a packet from the source to the destination. Assuming that packets select their SD pair uniformly from $\mathscr{S}$ we define the *average* and *worst-case* delay, respectively, as

$$\mathbf{E}[D] = (\textstyle\sum_{s\in\mathscr{S}} D_s)/|\mathscr{S}|,$$

$$D_{\max} = \max_{s\in\mathscr{S}} D_s.$$

Theorem IV.1. *[Delay in CFR] Let λ and λ' denote the maximum output rate from and maximum input rate to any arbitrary node in the network, respectively, i.e.,*

$$\lambda = \max_i\{\textstyle\sum_{j=1}^{N}\lambda(i,j)\},\quad \lambda' = \max_j\{\textstyle\sum_{i=1}^{N}\lambda(i,j)\}. \tag{7}$$

Let also $S_{\max} = \max_i|\mathscr{N}(i)|$ and $S_{\min} = \min_i|\mathscr{N}(i)|$ denote the largest and smallest number of nodes assigned to an MF, respectively. Finally, let L be the length of the longest MF cycle. If

$$\frac{N\min(\lambda,\lambda')}{W_2} + \frac{S_{\max}(\lambda+\lambda')}{W_1} < 1,$$

then

$$D_{max} \le 3T^C_{\max},$$
$$\text{where } T^C_{\max} = \frac{L}{V(1-\frac{N\min(\lambda,\lambda')}{W_2}-\frac{S_{\max}(\lambda+\lambda')}{W_1})}, \tag{8}$$

and

$$\mathbf{E}[D] \le \min\left\{\frac{M\binom{S_{\max}}{2}(\frac{3T^C_{\max}}{2}) + (\binom{N}{2}-M\Gamma)(\frac{5T^C_{\max}}{2})}{\binom{N}{2}},\right.$$
$$\left.\frac{5T^C_{\max}}{2}\right\},\ \text{where } \Gamma = \begin{cases}\binom{S_{\min}}{2}, & \text{if } S_{\min}\ge 2,\\ 0, & \text{otherwise.}\end{cases} \tag{9}$$

Proof: From Eq. (2) we have

$$Y = \frac{T}{W_2}\sum_{k=1}^{M}\sum_{l\in\mathscr{N}(k)}\sum_{k'=1,k'\neq k}^{M}\sum_{l'\in\mathscr{N}(k')}\lambda(l,l')$$
$$\leq \frac{T}{W_2}\sum_{k=1}^{M}\sum_{l\in\mathscr{N}(k)}\lambda = \frac{T}{W_2}N\lambda.$$

From Eq. (2) we also can write

$$Y \leq \frac{T}{W_2}\sum_{k'=1}^{M}\sum_{l'\in\mathscr{N}(k')}\sum_{k=1}^{M}\sum_{l\in\mathscr{N}(k)}\lambda(l,l')$$
$$\leq \frac{T}{W_2}\sum_{k'=1}^{M}\sum_{l'\in\mathscr{N}(k')}\lambda' = \frac{T}{W_2}N\lambda'.$$

Hence,

$$\frac{Y}{T} \leq \frac{N\min(\lambda,\lambda')}{W_2}.$$

Similarly, from Eq. (3) we have

$$\frac{X_i^d}{T} \leq \frac{S_{\max}\lambda}{W_1}, \qquad \frac{X_i^u}{T} \leq \frac{S_{\max}\lambda'}{W_1}. \tag{10}$$

Finally, using Eq. (6) and the inequalities above, we have

$$T \leq \frac{L}{V(1-\frac{N\min(\lambda,\lambda')}{W_2}-\frac{S_{\max}(\lambda+\lambda')}{W_1})} \triangleq T^C_{\max}. \tag{11}$$

The data need to be relayed by at most one MF, so the maximum number of MFs that carry the data is 2. Each MF carries the data at most for T units of time, and the data waits at most for T units of time before it is collected. Therefore,

$$D_{\max} \leq 3T \leq 3T^C_{\max}.$$

The average waiting time for the data before it is picked up is $T/2$ and if the destination is within the same cycle, then the maximum additional time to deliver the data is less than T. However, if the destination is in a different cycle, then the average delay is less than $\frac{3T}{2}+T$. The former case includes at most $M\binom{S_{\max}}{2}$ SD pairs and the latter one includes at most $\binom{N}{2}-M\Gamma$ SD pairs. Also, there are a total of $\binom{N}{2}$ SD pairs in the network. This implies

$$E[D] \leq \frac{M\binom{S_{\max}}{2}(\frac{3T}{2})+(\binom{N}{2}-M\Gamma)(\frac{5T}{2})}{\binom{N}{2}}, \tag{12}$$

where Γ is as defined in (9). Since the total number of SD pairs is $\binom{N}{2}$ and the average delay does not exceed $5T/2$ no matter where the destination is, we also have $E[D] \leq (5T/2)$. Combining this with (12) and (11) yields (9). ■

We note that the worst case bounds on Y, X_i^d, and X_i^u we derived are tight if the traffic is "load-balanced," that is, $\sum_{j=1}^{N}\lambda(i,j)$ is equal for all i or $\sum_{i=1}^{N}\lambda(i,j)$ is equal for all j (cf. Eq. (7)). Similarly, the inequality $D_{\max} \leq 3T$ is also tight assuming no MF coordination takes place.

The following Corollary follows from the fact that all upper bounds we derived in the proof above are linear in the data rates and L.

Corollary IV.2. *Suppose that the data rates between the nodes are random variables and the maximum output rate from and maximum input rate to any arbitrary node, denoted by $\tilde{\lambda}$ and $\tilde{\lambda}'$ (cf. (7)), respectively, satisfy*

$$\tilde{\lambda} \leq \lambda, \quad \tilde{\lambda}' \leq \lambda' \qquad \textit{w.p.1.}$$

Assume further that the maximum length, $\tilde{L}$, of the MF cycles is also stochastic such that $\tilde{L} \leq L$, w.p.1. Then the bound (8) on the worst-case delay holds w.p.1.

B. Scalability

Scalability is an important metric when designing a routing protocol. There has been a comprehensive body of work on the capacity of wireless networks in scenarios where each node wishes to send data to at most one destination. [21] considered a network of N randomly deployed nodes in a unit area disc where sources and destinations are chosen randomly. The throughput per node was found to be $\Theta(\frac{1}{N\log N})$. [22] assumed all nodes to be randomly mobile within a unit disc area and with data traveling over only two hops the throughput was found to be $\Theta(1)$. However, this improvement comes at the cost of increased delay. A series of papers followed that tried to characterize the delay/capacity relationship as a function of node mobility (e.g., [23], [24], [25]).

Another scenario for a network with mobile nodes was considered in [26]. The network consisted of N static nodes acting as sources and destinations for data and M randomly mobile nodes used as relays. Using a routing scheme proposed in [26], the throughout was found to be $O(\frac{M}{N\log^3 N})$ with an average delay of $O(1)$. A similar model was considered in [16] with controlled mobility. [16] showed that if mobility is fully under our control, then for $M=\alpha N$ where α is a constant such that $0<\alpha\leq 1$, a per node throughput of $O(1)$ with delay of $O(1)$ is achievable for scenarios where each node has only one destination. In the sequel, we extend this result to the multi-destination case. The key is the full cooperation of MFs to relay the data.

Our results in this section rely on bounding the inter-MF relay time. We account for "primary" interference constraints on MF transmissions. In particular, data exchanges are pairwise and an MF can not be communicating with more than one other MF at a time. We do not however account for secondary interference whereby two transmissions between two separate MF pairs may interfere if they use the same frequency and occur in close proximity. We assume that such interference can be handled by lower-level Media Access Control (MAC) protocols. Note that such secondary interference can be eliminated if interfering transmissions use different frequencies or a code division multiple access scheme. Even if scheduling (time division) is used, this would imply some additional delay which can be accounted for by some multiplicative factor to the expression we obtain. It should be apparent that our focus is to characterize the order of magnitude of inter-MF relay time rather than obtaining an exact expression.

Our analysis leverages graph-theoretic results. Define a (simple) undirected graph $\mathscr{G}=(\mathscr{V},\mathscr{E})$ where the nodes $\mathscr{V}=\{1,\ldots,M\}$ correspond to the MFs and introduce an edge between any two nodes that have to communicate. Assign to each edge a weight representing the data communication requirements. A *matching* is a set of non-adjacent edges (i.e., no two edges in the matching are incident to the same node). A *perfect matching* is a matching where every node of the graph is incident to exactly one edge of the matching. An *edge-coloring* is a partition of $\mathscr{E}$ into matchings, where each matching is assigned a different color. A k-edge-coloring is an edge coloring with k colors. The smallest k for which a k-edge-coloring exists is the *edge-coloring number* of $\mathscr{G}$ which we will denote by $\chi'(\mathscr{G})$. Let also $\Delta(\mathscr{G})$ denote the maximum node degree. Vizing has shown (see [27, Chap. 28]) the following result for (simple) graphs: $\Delta(\mathscr{G}) \leq \chi'(\mathscr{G}) \leq \Delta(\mathscr{G})+1$.

Lemma IV.3. *Suppose that $\mathscr{G}$ is a complete graph. Then $\chi'(\mathscr{G}) = M-1$ if M is even and $\chi'(\mathscr{G}) = M$ if M is odd.*

Proof: Since $\Delta(\mathscr{G}) = M-1$, from Vizing's result it follows that $M-1 \leq \chi'(\mathscr{G}) \leq M$. Note that by assigning color $i+j \mod M$ to each edge (i,j) we obtain a valid edge-coloring. Consider first the case of M being odd. Suppose that there exists an $(M-1)$-edge-coloring of $\mathscr{G}$. Then, $\mathscr{E}$ can be partitioned into $M-1$ disjoint matchings $\mathscr{E}_1, \ldots, \mathscr{E}_{M-1}$ such that $\mathscr{E} = \cup_{i=1}^{M-1} \mathscr{E}_i$. Since M is odd, each of these matchings leaves an unmatched node and requires at most $(M-1)/2$ edges. Hence, $|\mathscr{E}| \leq (M-1)^2/2$. On the other hand, $\mathscr{G}$ is a complete graph and $|\mathscr{E}| = M(M-1)/2 > (M-1)^2/2$, which is a contradiction. We conclude that there is no $(M-1)$-edge-coloring of $\mathscr{G}$ and $\chi'(\mathscr{G}) = M$ when M is odd.

Next consider the case where M is even. Form the subgraph, denoted by $\tilde{G} = (\tilde{V}, \tilde{E})$, by removing node M from $\mathscr{G}$. $\tilde{G}$ is a complete graph with $M-1$ nodes (an odd number). We have already shown that the edge-coloring number of $\tilde{G}$ is $M-1$ and we can color the edges of $\tilde{G}$ by assigning color $i+j \mod (M-1)$ to each edge (i,j). Note that the color $c_i = 2i \mod (M-1)$ is not used to color an edge incident to node i, for each $i = 1, \ldots, M-1$. Since $M-1$ is odd, $i \neq j$ implies that $c_i \neq c_j$. Thus, we can color $\mathscr{G}$ by using this coloring of $\tilde{G}$ and using color c_i on edge (i, M) for each $i = 1, \ldots, M-1$. This yields an $(M-1)$-edge coloring of $\mathscr{G}$. ■

Note that the proof above is constructive and specifies how exactly to color the edges of $\mathscr{G}$. Before we proceed, we state the following theorem from [28] which gives a tight upper bound on the length of the worst shortest Hamiltonian cycle.

Theorem IV.4. *([28]) $\mathscr{V}_N$ denotes a set of N points in $[0,1]^2$, $\mathscr{A}$ denotes a subset of the edges of the complete graph defined on the points of V_N, and $|e|$ denotes the Euclidean distance between two points connected by edge $e \in \mathscr{A}$. Let*

$$\rho_{TSP}(N) = \max_{\mathscr{V}_N} \left\{ \min_{T} \textstyle\sum_{e \in T} |e| : T \text{ is a tour of } \mathscr{V}_N \right\}.$$

Then $\rho_{TSP}(N) \leq \sqrt{2N} + \frac{7}{4}$.

Theorem IV.5. *Assume N nodes are placed arbitrarily on a unit square, each of which is transmitting with a total rate λ to all other nodes such that the data rate between any pair of nodes is the same. If*

$$\lambda < \frac{1}{2(\frac{N^2}{M(N-1)W_2} + \frac{N}{MW_1})}, \tag{13}$$

then there exists a CFR scheme for which

$$D_{max} \leq \frac{3(\sqrt{2(\frac{N}{M}+1)} + \frac{7}{4})}{V[1 - 2\lambda(\frac{N^2}{M(N-1)W_2} + \frac{N}{MW_1})]}, \tag{14}$$

$$\mathbf{E}[D] \leq \frac{N(\frac{5}{2} - \frac{1}{M}) - \frac{3}{2}}{N-1} \frac{(\sqrt{2(\frac{N}{M}+1)} + \frac{7}{4})}{V[1 - 2\lambda(\frac{N^2}{M(N-1)W_2} + \frac{N}{MW_1})]}. \tag{15}$$

Proof: Consider a configuration of the CFR scheme where we assign N/M (assumed integer) nodes to each MF. Using Eq. (10) we have

$$\frac{X_i^d}{T} \leq \frac{(\frac{N}{M})\lambda}{W_1}, \qquad \frac{X_i^u}{T} \leq \frac{(\frac{N}{M})\lambda}{W_1}.$$

To bound the inter-MF relay time we use Lemma IV.3. Specifically, the edge-coloring of Lemma IV.3 partitions the MF communications into a number of matchings, M if M is odd and $M-1$ if M is even. Each matching corresponds to inter-MF communications that can proceed simultaneously; let us call a "stage" the process of completing all these inter-MF relays that correspond to a matching. Since there are N/M nodes assigned to each MF, every stage includes traffic from $(N/M)^2$ pairs of nodes which needs to be relayed bidirectionally. Thus, every stage takes no more than $(2T\frac{N}{M}\frac{N}{M}\frac{\lambda}{N-1})/W_2$ units of time. Since the number of matchings is either M, if M is odd, or $M-1$, if M is even, we obtain

$$\frac{Y}{T} \leq \frac{M2\frac{N}{M}\frac{N}{M}\frac{\lambda}{N-1}}{W_2}.$$

Using Eq. (6) and Theorem IV.4, we have

$$T \leq \frac{(\sqrt{2(\frac{N}{M}+1)} + \frac{7}{4})}{V[1 - 2\lambda(\frac{N^2}{M(N-1)W_2} + \frac{N}{MW_1})]}.$$

Note that for λ to be feasible, we need (13). Using now the same steps that led to the bounds on $D_{\max}$ and $\mathbf{E}[D]$ in the proof of Thm. IV.1, and some algebra, we obtain the result. ■

Theorem IV.6. *[Scalability of CFR] Consider the setup of Thm. (IV.5). If $M = \alpha N$ for some positive constant α, then constant (in M and N) per node throughput with constant worst-case delay is achievable using the CFR scheme.*

Proof: Substituting $M = \alpha N$ in (14), we obtain that the worst-case delay is upper-bounded while the constant per-node throughput $\lambda < \frac{\alpha}{2(\frac{2}{W_2} + \frac{1}{W_1})}$ is feasible. ■

Note that the approach used in [16] for deriving the upper-bound on delay, if applied to the case of Theorem IV.6, would not achieve constant delay. The following result from [29] stated in [30] clarifies this point.

Theorem IV.7. *(From [29] in [30]) Consider the setup of Thm. IV.4. It follows $\rho_{TSP}(N) \geq 2(12)^{-\frac{1}{4}}\sqrt{N}$.*

For scenarios where each node has only one destination, [16] suggested to group the nodes into M groups and assign each group to a different MF. The MF would be responsible for picking up the data and delivering directly to the corresponding destination without relays. For constant per-node throughput, [16] showed that the delay remains constant as the number of nodes and MFs grow. However, for scenarios where each node wishes to transmit to all other nodes, the approach in [16] can not yield a constant delay. To that end, notice that if each MF is assigned to a fraction of the source nodes (say N/M), then as N grows, the length of the path that each MF should travel in order to visit all destinations of those N/M source nodes would grow to infinity (from Theorem IV.7). Since the velocity of MFs, V, is bounded, the delay increases as N grows. In contrast, our CFR scheme can achieve $O(1)$ per node throughput with $O(1)$ worst-case delay.

C. CFR Optimization

CFR optimization is defined as the problem of finding the optimal configuration of the CFR scheme (assigning nodes to MFs and determining a route for each MF) such that T in Eq. (6) is minimized. The CFR optimization is a hard problem. In particular, as W_1 and W_2 grow large, the inter-MF relay times and MF-node transfer times converge to zero, and for any assignment of nodes to MFs Eqs. (6) and (5) imply that the problem reduces to finding the shortest route for each MF. This amounts to solving M *Traveling Salesman Problems (TSPs)*, thus establishing that CFR optimization is NP-hard. To cope with the difficulty, we introduce a heuristic algorithm. We group nodes into M groups. Intuitively, a good assignment (grouping) is the one that leads to groups with minimal inter-group communications but also minimal distances between nodes in the same group. The former minimizes inter-MF relay times and the latter MF travel times.

At first, a set $\mathscr{S}$ of M nodes is selected such that $\sum_{i\in\mathscr{S}}\sum_{j\in\mathscr{S},j\neq i}\lambda(i,j)$ is minimized. Each of these M nodes initializes a group for each MF. To perform this step, we suggest using a greedy method: start with an arbitrary node and keep adding one node at a time so that the node to be added has the smallest *cumulative rate* to the nodes already selected (the cumulative rate of a node to a group of nodes is the sum of rates between the node and all individual nodes within the group). If a tie occurs, we choose a node that has the smallest cumulative rate to all other nodes in the network. The proposed greedy method takes $O(M)$ amount of work to be completed.

Once this group initialization phase is over we continue by assigning the remaining $(N-M)$ nodes to the M groups, one at a time. For each node we assign to a group, the objective is to minimize the resulting cycle time $T=\max_i\{T_i\}$, where the T_i's, $i=1,\ldots,M$, solve the following set of equations:

$$T_i = \frac{\tilde{L}_i}{V} + \left\{\frac{T_i}{W_1}\sum_{k=1}^{M}\sum_{l\in\mathscr{S}(i)}\sum_{l'\in\mathscr{S}(k)}\lambda(l,l')\right\} + \left\{\frac{T_i}{W_2}\sum_{k=1}^{M}\sum_{k'=1,k'\neq k}^{M}\sum_{l\in\mathscr{S}(k)}\sum_{l'\in\mathscr{S}(k')}\lambda(l,l')\right\} + \left\{\frac{T_i}{W_1}\sum_{k=1}^{M}\sum_{l\in\mathscr{S}(k)}\sum_{l'\in\mathscr{S}(i)}\lambda(l,l')\right\},\ \forall i. \qquad (16)$$

In the above, $\tilde{L}_i$ is the length of MF i's route and $\mathscr{S}(i)$ is the set of nodes that have already been assigned to MF i plus the node being considered. When adding nodes to a group, we use a *cheapest insertion* method to form a new Hamiltonian cycle for the MF assigned to the group. That is, the new node is inserted between two consecutive nodes in the current cycle so that the smallest increase in length is observed. We consider adding the new node to each one of the M groups and for each case we compute the corresponding T_i's from (16) and set $T=\max_i\{T_i\}$. We then decide to add the node to some group such that it results in minimal T. The assignment of the $(N-M)$ nodes to the M groups takes $O(N^2M)$ amount of work.

After all nodes are assigned, we calculate each MF's cycle time T_i using Eq. (1). The total cycle time T can also be calculated using Eq. (6). Then, the MF with largest T_i is recognized. We call this MF the "critical MF" and its group the "critical group." We target the critical group and try to shorten the critical MF's cycle time by de-assigning one of its nodes and assigning it to another MF. If due to this node-exchange a reduction in total cycle time is observed, then we apply the exchange. We continue this process until no further improvement is found. This solution always achieves the average and the worst-case delays given in Theorem IV.1. To limit the amount of work for this improvement phase we can limit the number of iterations by $O(N^2M)$ so that the whole CFR optimization heuristic we described maintains an $O(N^2M)$ complexity.

V. Bus Ferry Relaying: BFR

In this section, we propose our second scheme. Fig. 2 shows an example illustrating the BFR scheme with a single bus. In the BFR scheme with a single bus, N nodes are assigned to $(M-1)$ MFs and data is relayed among MFs via the remaining MF which is called *the bus*. Each of those $(M-1)$ MFs collects and delivers data within its own group (the nodes assigned to that MF), and the bus circulates the data among groups. No direct relaying occurs among those $(M-1)$ MFs. The bus meets with each MF at a contact point exactly once in each cycle. We choose one of the nodes assigned to MF i as the contact point between the bus and MF i. Each MF, after exchanging data with the bus at its contact point, travels along its route (which is a Hamiltonian cycle including all nodes assigned to it), exchanging data with its nodes.

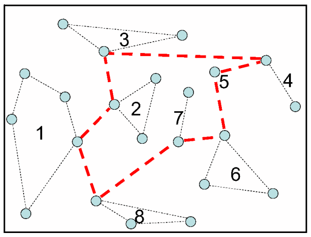


Fig. 2. An example illustrating BFR scheme - the blue circles represent the nodes, the black lines show the routes of eight MFs, and the red dashed line shows the route of the bus.

A. Average and worst case delay

We refer to the time MF i takes to complete these activities as MF i's cycle time, we denote it by T_i, and we refer to the time the bus takes to finish one cycle as the *bus cycle time* T_{bus}. The cycle time of MF i consists of the following terms (we use the same notation introduced in earlier sections):

- *MF-bus relay time* (Y_i): the transfer time (Y_i^u) MF i needs in order to relay to the bus all the buffered data it has collected from its nodes, plus the transfer time (Y_i^d) the bus needs in order to relay to MF i all the data that have been relayed to it from other MFs. We have
$$Y_i = Y_i^u + Y_i^d, \quad i = 1, 2, \ldots, (M-1). \tag{17}$$
- *MF-node transfer time* (X_i): the transfer time (X_i^d) MF i needs to collect all the data from its nodes, plus the transfer time (X_i^u) MF i needs to deliver all its buffered data to its nodes. We have
$$X_i = X_i^d + X_i^u \quad i = 1, 2, \ldots, (M-1). \tag{18}$$
- *MF travel time* (A_i): the travel time of MF i.

The cycle time of the bus T_{bus} consists of:

- *Relay time* (Y_{bus}): the transfer time that all $(M-1)$ MFs need in order to relay to the bus all their buffered data plus the transfer time that bus needs in order to relay to the MFs all its buffered data. We have
$$Y_{\text{bus}} = \textstyle\sum_{i=1}^{M-1} Y_i. \tag{19}$$
- *Bus travel time* (A_{bus}): the travel time of the bus.

Since MFs do not exchange data directly among themselves, synchronization among MFs is not required. However, synchronization between the MFs and the bus is required. To that end, the cycle time of the system, T, can be chosen as the maximum of the MFs' cycle times and the bus cycle time. Next, we calculate the cycle time for a given configuration of the BFR scheme. We have

$$T_i = Y_i + X_i + A_i, \ \forall i, \ T_{\text{bus}} = (\textstyle\sum_{i=1}^{M-1} Y_i) + A_{\text{bus}},$$

where

$$Y_i^u = \frac{T}{W_2} \sum_{k=1, k\neq i}^{M} \sum_{l \in \mathcal{N}(k)} \sum_{l' \in \mathcal{N}(i)} \lambda(l', l), \tag{20}$$

$$Y_i^d = \frac{T}{W_2} \sum_{k=1, k\neq i}^{M} \sum_{l \in \mathcal{N}(k)} \sum_{l' \in \mathcal{N}(i)} \lambda(l, l'), \tag{21}$$

$$X_i^d = \frac{T}{W_1} \sum_{k=1}^{M} \sum_{l \in \mathcal{N}(i)} \sum_{l' \in \mathcal{N}(k)} \lambda(l, l'), \tag{22}$$

$$X_i^u = \frac{T}{W_1} \sum_{k=1}^{M} \sum_{l \in \mathcal{N}(k)} \sum_{l' \in \mathcal{N}(i)} \lambda(l, l'), \tag{23}$$

$$A_i = L_i/V, \qquad A_{\text{bus}} = L_{\text{bus}}/V. \tag{24}$$

In the equations above, L_i is the length of the route of MF i, and L_{bus} is the length of the route of the bus. Note that if $T = T_i$ then

$$T_i = \frac{A_i}{1 - \frac{Y_i}{T} - \frac{X_i}{T}}, \tag{25}$$

and if $T = T_{\text{bus}}$ then

$$T_{\text{bus}} = \frac{A_{\text{bus}}}{1 - \sum_{i=1}^{M-1} \frac{Y_i}{T}}. \tag{26}$$

T is therefore the maximum of these quantities and it follows

$$T = \max(\max_i \{T_i\}, T_{\text{bus}}). \tag{27}$$

Note that $\frac{Y_i}{T}$ and $\frac{X_i}{T}$ do not depend on T.

We next derive bounds on the maximum and average delay. Let, as in Sec. IV, $S_{\max}$ and $S_{\min}$ denote the largest and smallest number of nodes assigned to an MF, respectively. Each MF circulates over a Hamiltonian cycle including all its nodes. Let L the length of the longest such cycle among the $M-1$ MFs. Let also L' denote the length of the bus route.

Theorem V.1. *[BFR delay] Let λ and λ' be defined as in (7). If*

$$\left(\frac{1}{W_2} + \frac{1}{W_1}\right) < \frac{1}{S_{\max}(\lambda + \lambda')}, \qquad W_2 > 2N \min(\lambda, \lambda'),$$

then there exists a BFR scheme for which

$$D_{max} \le 4T_{\max}^B, \ \textit{where}\ T_{\max}^B = \frac{1}{V} \max \left\{ \frac{L'}{1 - \frac{2N \min(\lambda, \lambda')}{W_2}}, \right. \\ \left. \frac{L}{1 - S_{\max}(\lambda + \lambda')(\frac{1}{W_2} + \frac{1}{W_1})} \right\}, \tag{28}$$

and

$$\mathbf{E}[D] \le \min \left\{ \frac{7T_{\max}^B}{2}, \right. \\ \left. \frac{(M-1)\binom{S_{\max}}{2}(\frac{3T_{\max}^B}{2}) + (\binom{N}{2} - (M-1)\Gamma)(\frac{7T_{\max}^B}{2})}{\binom{N}{2}} \right\}, \tag{29}$$

where Γ is as defined in (9).

Proof: Since the cycle of each MF has at most $S_{\max}$ nodes, using Eqs. (17), (20) and (21) we have

$$\frac{Y_i}{T} \le \frac{S_{\max}(\lambda + \lambda')}{W_2}.$$

Similarly, using Eqs. (18), (22) and (23) we obtain

$$\frac{X_i}{T} \le \frac{S_{\max}(\lambda + \lambda')}{W_1}.$$

Using Eqs. (20) and (21) we have

$$\textstyle\sum_{i=1}^{M-1} (Y_i^u/T) \le N\lambda/W_2, \quad \sum_{i=1}^{M-1} (Y_i^d/T) \le N\lambda'/W_2.$$

Since the data to be downloaded to MFs (from the bus) and uploaded to the bus (from the MFs) is exactly the same and using Eq. (17) it follows that

$$\textstyle\sum_{i=1}^{M-1} (Y_i/T) \le (2N \min(\lambda, \lambda'))/W_2.$$

Finally, using Eq. (27), we can write

$$T \le \frac{1}{V} \max \left(\frac{L}{1 - S_{\max}(\lambda + \lambda')(\frac{1}{W_2} + \frac{1}{W_1})}, \right.$$

$$\left.\frac{L'}{1-\frac{2N\min(\lambda,\lambda')}{W_2}}\right) \triangleq T^B_{\max}.$$

The destination of the data belongs to either the same cycle that the source belongs to, or to another cycle. In the former case, the data reaches the destination within at most $2T$ time units. In the latter case, the data needs to be relayed twice (once from the MF serving the source node to the bus, and once from the bus to the MF serving the destination node). Moreover, it reaches the destination within $4T$ time units since the data is carried by the source MF, the bus, and the destination MF, where each MF carries the data for at most T time units and the data waits at most for T time units before it is collected. Therefore

$$D_{\max} \leq 4T \leq 4T^B_{\max}.$$

The average waiting time of data before it is picked up is $T/2$ and if the destination of the data is within the origin segment, then the maximum time to deliver the data after it is picked up would be less than T. However, if the destination of the data is in a segment other than the origin, the average delay is less than $\frac{3T}{2}+2T$. The former case includes at most $(M-1)\binom{S_{\max}}{2}$ SD pairs and the latter one includes at most $\binom{N}{2}-(M-1)\Gamma$ SD pairs, where Γ is as defined in (9). Also, there are totally $\binom{N}{2}$ SD pairs in the network. Therefore, after some algebra we obtain

$$\mathbf{E}[D] \leq \frac{(M-1)\binom{S_{\max}}{2}(\frac{3T^B_{\max}}{2}) + (\binom{N}{2}-(M-1)\Gamma)(\frac{7T^B_{\max}}{2})}{\binom{N}{2}}. \quad (30)$$

Since the total number of SD pairs is $\binom{N}{2}$ and the average delay does not exceed $7T^B_{\max}/2$ no matter where the destination is, we also have $E[D] \leq (7T^B_{\max}/2)$. Combining this with (30) yields (29). ■

We remark that a result analogous to Corollary IV.2 can be obtained for Thm. V.1 as well. Using Thm. IV.4 we obtain the following Corollary.

Corollary V.2. *Assume N nodes are placed arbitrarily on a unit square, each of which transmitting with rate λ to other nodes (such that the data rate between any pair of nodes is the same). If*

$$\left(\frac{1}{W_2}+\frac{1}{W_1}\right) < \frac{1}{2S_{\max}\lambda} \quad \text{and} \quad W_2 > 2N\lambda,$$

then there exists a BFR scheme for which (28) and (29) hold with $T^B_{\max}$ replaced by

$$\tilde{T}_{up} = \frac{1}{V}\max\left\{\frac{\sqrt{2(M-1)}+\frac{7}{4}}{1-\frac{2N\lambda}{W_2}}, \frac{\sqrt{2S_{\max}}+\frac{7}{4}}{1-2S_{\max}\lambda(\frac{1}{W_2}+\frac{1}{W_1})}\right\}.$$

B. BFR Optimization

BFR Optimization is defined as the problem of finding the optimal configuration of the BFR scheme such that T in Eq. (27) is minimized. Using the same argument as in Sec. IV-C it can be shown that it is NP-hard (reduction from TSP). We resort to a heuristic algorithm. We will group nodes into $(M-1)$ groups. We need to assign the nodes to MFs such that, on one hand, it leads to small inter-group communications, and, on the other hand, nodes geographically close to each other are assigned to the same group. In addition, we need to consider the length of the bus route so that the bus cycle time does not exceed $\max_i\{T_i\}$. Our heuristic algorithm for the BFR scheme has two phases. In the first phase, we assign nodes to MFs, and in the second phase, we establish the bus route and attempt to further minimize delay.

1) Phase I (Assigning nodes to MFs): At first, a set $\mathscr{S}$ of $(M-1)$ nodes out of N are chosen such that $\sum_{i\in\mathscr{S}}\sum_{j\in\mathscr{S},j\neq i}\lambda(i,j)$ is minimized. Each of these $M-1$ nodes initializes a group assigned to an MF to which we will add nodes later. This initial step can be executed by using a greedy method: start with an arbitrary node and add additional nodes one at a time so that the node to be added has the smallest cumulative rate (defined as in Sec. IV-C) with the nodes already selected.

Once this group initialization phase is over we continue by assigning the remaining $(N-M+1)$ nodes to these groups, one at a time. With each node we add to a group, the objective is to minimize the cycle time $T=\max_i\{T_i\}$, where the T_i's for $i=1,\ldots,(M-1)$ are solutions to the following set of equations:

$$\begin{aligned} T_i = &\frac{\tilde{L}_i}{V} + \left\{\frac{T_i}{W_1}\sum_{k=1}^{M-1}\sum_{l\in\mathscr{S}(i)}\sum_{l'\in\mathscr{S}(k)}\lambda(l,l')\right\} \\ &+ \left\{\frac{T_i}{W_2}\sum_{k=1}^{M-1}\sum_{k'=1,k'\neq k}^{M-1}\sum_{l\in\mathscr{S}(k)}\sum_{l'\in\mathscr{S}(k')}\lambda(l,l')\right\} \\ &+ \left\{\frac{T_i}{W_1}\sum_{k=1}^{M-1}\sum_{l\in\mathscr{S}(k)}\sum_{l'\in\mathscr{S}(i)}\lambda(l,l')\right\}. \end{aligned} \quad (31)$$

In the above, $\tilde{L}_i$ is the length of MF i's route and $\mathscr{S}(i)$ is the set of nodes that have already been assigned to MF i plus the node being considered. Note that when adding nodes to a group, we also consider re-optimizing the route of that MF in order to make sure that the route has minimum length. This can be done by using a *cheapest insertion method*: the new node is inserted between two consecutive nodes in the current cycle so that the smallest increase in cycle length is observed. For each unassigned node, we consider adding it to each one of the $M-1$ groups. In each case we compute the corresponding T_i from (31) and set $T=\max_i\{T_i\}$. We then add the node to the group that results in minimal T. Using similar analysis as in the CFR case, Phase I has a complexity of $O(N^2M)$.

2) Phase II (Establishing the bus route): The bus route is a Hamiltonian cycle including $(M-1)$ nodes in a way that exactly one node from each of the $(M-1)$ groups is visited. We have the freedom to select which node from each group will be visited and, in addition, we want the bus route to have minimum length so that the bus travel time is minimized. This problem is a well-studied problem and is called the *Generalized Traveling Salesman Problem* (GTSP) in the literature. Several approaches transform the GTSP into the classical TSP [31], [32], [33], [34], [35]. The generalized 2-opt method [36] can also be used. Such a method has no complexity bounds on the number of iterations needed for convergence but one can bound the iterations performed to maintain low complexity. Having established the bus route, we then calculate each MF's cycle time T_i as well as the bus cycle time T_{bus} using Eqs. (25) and (26). The total cycle time T can also be calculated using Eq. (27). At each step, one of the following conditions applies:

- $T_{\text{bus}} < \max_i\{T_i\}$: Then the MF with the largest T_i is recognized. We call this MF the "critical MF" and the critical MF's group the "critical group." We target the critical group and we try to shorten its cycle time by de-assigning one of its nodes and assigning it to another MF (e.g., by a cheapest insertion method). If due to this node-exchange a reduction in total cycle time is observed, then we apply the exchange and we repeat the process.
- $T_{\text{bus}} \geq \max_i\{T_i\}$: Then the bus cycle time would be targeted. In this case, we find and de-assign the nodes with minimal cumulative rate to the rest of the nodes and assign them to the group which results in the largest decrease in T_{bus} (again by cheapest insertion). If this node-exchange results in a reduction in total cycle time, then we apply the exchange and we repeat the process.

We continue this phase until no further improvement is found. Note that this solution always achieves the average and worst-case delays given in Theorem V.1. As in the CFR case we can limit the number of iterations of the improvement phase in order to maintain low complexity.

We close this section by noting that the BFR scheme, as presented, may not scale well as the number of MFs increases because the bus has to make many "stops." One way of making BFR scalable is to use multiple buses, decomposing the overall coverage area into sections and assigning one bus to each section. Buses can then meet at section boundaries (e.g., using an NFR-like scheme we present in the next section) to relay messages among them.

VI. Neighbor Ferry Relaying: NFR

The idea behind our last scheme, NFR, is to form a Hamiltonian cycle which includes all nodes and then assign each MF to a contiguous segment of this cycle –called the *path*– in a way that each node in the path is assigned to a single MF (see Fig. 3). Without loss of generality, we start from an arbitrary point in the cycle and move in the forward direction (let us define clockwise and counterclockwise directions as "forward" and "backward" directions, respectively). As we move forward, we assign the first $n(1)$ nodes to MF 1, the next $n(2)$ nodes to MF 2, and so on and so forth until we arrive back to the start point. We denote the set of assigned nodes to MF i by $\mathscr{N}(i)$.

MFs assigned to each path work serially: each MF relays its buffered data only to its two adjacent MFs, and data circulate (via multiple relays) in order to reach their destinations. For efficiency reasons, the scheme requires the data to be relayed only along the direction (either forward or backward) which is closer to the destination (in case either direction results in the same delay, we assume the data is circulated in the forward direction). Each MF moves forward and backward periodically between the two contact points located at each end of its assigned path. We refer to the time that is taken by MF i to complete its forward and backward trips as *forward time* (T_i^F) and *backward time* (T_i^B) of MF i, respectively. We also refer to the two adjacent MFs that meet with MF i at contact points P^i and P^{i+1} (as shown in Fig. 3) as MF i's *backward neighbor* and *forward neighbor*, respectively.

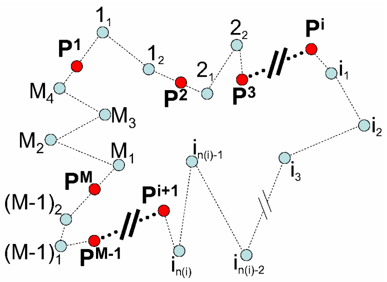


Fig. 3. An example illustrating the NFR scheme. Nodes $i_1, \ldots, i_{n(i)}$ denote the nodes assigned to MF i. Points P_i in the line segments connecting neighboring paths are contact points where the MFs meet to exchange data.

A. Average and worst case delay

The forward time of MF i, T_i^F, consists of the following terms:

- *Inter-MF relay time* (Y_i^F): the transfer time (Y_i^{Fu}) that MF i needs in order to relay to its forward neighbor all the buffered data that have been relayed to MF i from its backward neighbor during their last meeting at contact point P^i as well as all the buffered data that MF i has collected from all its own nodes (when last traveling from P^i to P^{i+1}) that need to be relayed to MF i's forward neighbor, plus the transfer time (Y_i^{Fd}) that MF i's forward neighbor needs in order to relay to MF i all its buffered data. We have

$$Y_i^F = Y_i^{Fu} + Y_i^{Fd}. \tag{32}$$

- *MF-node transfer time* (X_i^F): the transfer time (X_i^{Fu}) that MF i needs in order to deliver to its nodes all the data that have been relayed to it from its backward neighbor during their last meeting at contact point P^i and are destined to MF i's nodes, plus the transfer time (X_i^{Fd}) that MF i needs in order to collect all the data from its assigned nodes and need to be relayed to MF i's forward neighbor during their next meeting, plus the transfer time (X_i^{Fl}) that MF i needs in order to collect and deliver all the data from its nodes destined to the nodes located in MF i's own path and lying between the source node and contact point P^{i+1}. We have

$$X_i^F = X_i^{Fu} + X_i^{Fd} + X_i^{Fl}.$$

- *MF travel time* (A_i^F): the travel time of MF i from contact point P^i to P^{i+1}.

Collecting all these we have

$$T_i^F = Y_i^F + X_i^F + A_i^F.$$

The backward time of MF i, T_i^B, consists of the following terms:

- *Inter-MF relay time* (Y_i^B): the transfer time (Y_i^{Bu}) that MF i needs in order to relay to its backward neighbor all the buffered data that have been relayed to MF i from its forward neighbor during their last meeting at contact point P^{i+1}, as well as, all the buffered data that MF i has collected from all its own nodes (when last traveling from P^{i+1} to P^i) that need to be relayed to MF i's backward neighbor, plus the transfer time (Y_i^{Bd}) that MF i's backward neighbor needs in order to relay to MF i all its buffered data. We have
$$Y_i^B = Y_i^{Bu} + Y_i^{Bd}. \tag{33}$$
- *MF-node transfer time* (X_i^B): the transfer time (X_i^{Bu}) that MF i needs in order to deliver to its nodes all the data that have been relayed to it from its forward neighbor during their last meeting at contact point P^{i+1} and are destined to MF i's nodes plus the transfer time (X_i^{Bd}) that MF i needs in order to collect all the data that have originated and have been buffered in MF i's nodes and need to be relayed to MF i's backward neighbor during their next meeting, plus the transfer time (X_i^{Bl}) that MF i needs in order to collect and deliver all the data from its nodes destined to the nodes located in MF i's own path and lying between the source node and contact point P^i. We have
$$X_i^B = X_i^{Bu} + X_i^{Bd} + X_i^{Bl}.$$
- *MF travel time* (A_i^B): the travel time of MF i from contact point P^{i+1} to its other contact point P^i. We have
$$A_i^B = A_i^F.$$

Collecting the above we have

$$T_i^B = Y_i^B + X_i^B + A_i^B.$$

Since MFs exchange data directly, synchronization is required so that two adjacent MFs meet at the same time at their contact points. For this scheme to work, we require M to be even so that at each contact point two adjacent MFs are available to relay data to each other. Also to ensure that adjacent MFs are able to meet each other in every cycle, the scheme requires MFs in adjacent paths to move in reverse direction. In particular, and without loss of generality, assume that odd MFs start from their counter-clockwise contact point and even MFs from their clockwise contact point. So, initially, each pair of MFs starts from the same point. Odd MFs move clockwise and even MFs counter-clockwise. It can be seen that odd MFs will meet again with their backward neighbors at the initial point after a finite amount of time and no deadlock can occur. We define

$$T_i = 2\max\{T_i^F, T_i^B\} \tag{34}$$

and refer to it as MF i's *cycle time*. When the cycle times of two adjacent MFs are not the same, the contacts between MFs would occur irregularly, so the MF with shorter cycle time needs to wait for an additional period of time before its neighbor arrives and they can exchange data. It follows that the cycle time of the system can be chosen as the maximum of all MF cycle times: $T = \max_i\{T_i\}$. We will next proceed with characterizing delay and deriving bounds on average and worst-case delay.

Assume that a configuration of the NFR scheme is given (without loss of generality as presented in Fig. 3). To calculate the cycle time for a given configuration of the scheme, we define the function $\gamma(j,k)$ for $j \in \{1,2,\ldots,M\}$ and for $k \in \{-M,-M+1,\ldots,-1,1,\ldots,M-1,M\}$ which identifies the MF that is k segments away from MF j (in the forward direction if k is positive, and in the backward direction if k is negative) as follows:

$$\gamma(j,k) = \begin{cases} j+k, & \text{if } j+k \geq 1, \quad k<0, \\ M+j+k, & \text{if } j+k < 1, \quad k<0, \\ j+k, & \text{if } j+k \leq M, \quad k \geq 0, \\ -M+j+k, & \text{if } j+k > M, \quad k \geq 0. \end{cases}$$

We have

$$Y_i^{Fu} = \frac{T}{W_2} \sum_{k=0}^{\frac{M}{2}-1} \sum_{k'=1}^{\frac{M}{2}-k} \sum_{l \in \mathcal{N}(\gamma(i,-k))} \sum_{l' \in \mathcal{N}(\gamma(i,k'))} \lambda(l,l'), \tag{35}$$

$$Y_i^{Fd} = \frac{T}{W_2} \sum_{k=1}^{\frac{M}{2}-1} \sum_{k'=0}^{\frac{M}{2}-k-1} \sum_{l \in \mathcal{N}(\gamma(i,k))} \sum_{l' \in \mathcal{N}(\gamma(i,-k'))} \lambda(l,l'), \tag{36}$$

$$X_i^{Fu} = \frac{T}{W_1} \sum_{k=1}^{\frac{M}{2}} \sum_{l \in \mathcal{N}(\gamma(i,-k))} \sum_{l' \in \mathcal{N}(i)} \lambda(l,l'), \tag{37}$$

$$X_i^{Fd} = \frac{T}{W_1} \sum_{k=1}^{\frac{M}{2}} \sum_{l \in \mathcal{N}(\gamma(i,k))} \sum_{l' \in \mathcal{N}(i)} \lambda(l',l), \tag{38}$$

$$X_i^{Fl} = \frac{2T}{W_1} \sum_{l \in \mathcal{N}(i)} \sum_{l' \in \mathcal{N}_{F,l}(i)} \lambda(l,l'), \tag{39}$$

$$Y_i^{Bu} = \frac{T}{W_2} \sum_{k=0}^{\frac{M}{2}-2} \sum_{k'=1}^{\frac{M}{2}-k-1} \sum_{l \in \mathcal{N}(\gamma(i,k))} \sum_{l' \in \mathcal{N}(\gamma(i,-k'))} \lambda(l,l'), \tag{40}$$

$$Y_i^{Bd} = \frac{T}{W_2} \sum_{k=1}^{\frac{M}{2}} \sum_{k'=0}^{\frac{M}{2}-k} \sum_{l \in \mathcal{N}(\gamma(i,-k))} \sum_{l' \in \mathcal{N}(\gamma(i,k'))} \lambda(l,l'), \tag{41}$$

$$X_i^{Bu} = \frac{T}{W_1} \sum_{k=1}^{\frac{M}{2}-1} \sum_{l \in \mathcal{N}(\gamma(i,k))} \sum_{l' \in \mathcal{N}(i)} \lambda(l,l'), \tag{42}$$

$$X_i^{Bd} = \frac{T}{W_1} \sum_{k=1}^{\frac{M}{2}-1} \sum_{l \in \mathcal{N}(\gamma(i,-k))} \sum_{l' \in \mathcal{N}(i)} \lambda(l',l), \tag{43}$$

$$X_i^{Bl} = \frac{2T}{W_1} \sum_{l \in \mathcal{N}(i)} \sum_{l' \in \mathcal{N}_{B,l}(i)} \lambda(l,l'), \tag{44}$$

$$A_i^F = A_i^B = L_i/V. \tag{45}$$

In the equations above, $\mathscr{N}_{F,l}(i)$ denotes the set of nodes in MF i's path which are located after node l in the forward direction, $\mathscr{N}_{B,l}(i)$ is the set of nodes in MF i's path which are located after node l in the backward direction, and L_i is the length of MF i's path. Solving for T, we have

$$T = \max_i \left\{ \max\left(\frac{A_i^F}{\frac{1}{2} - \frac{Y_i^F}{T} - \frac{X_i^F}{T}}, \frac{A_i^B}{\frac{1}{2} - \frac{Y_i^B}{T} - \frac{X_i^B}{T}} \right) \right\}. \tag{46}$$

Note that $\frac{Y_i^F}{T}$, $\frac{X_i^F}{T}$, $\frac{Y_i^B}{T}$, and $\frac{X_i^B}{T}$ do not depend on T.

Theorem VI.1. *[Delay in NFR] Let λ and λ' be defined as in (7). Let also $S_{\max}$ denote the largest number of nodes assigned to an MF and L the length of the longest MF path. If*

$$\frac{M\min(\lambda,\lambda')}{W_2} + \frac{(\lambda+\lambda')}{W_1} < \frac{1}{2S_{\max}},$$

then there exists an NFR scheme for which

$$D_{max} \leq \left(\frac{M}{4} + \frac{3}{2}\right) T_{\max}^N, \text{ where}$$

$$T_{\max}^N = \frac{\frac{L}{V}}{\frac{1}{2} - S_{\max}\left[\frac{M\min(\lambda,\lambda')}{W_2} + \frac{\lambda+\lambda'}{W_1}\right]}. \tag{47}$$

and

$$\mathbf{E}[D] \leq \min\left\{ \frac{\left(\sum_{k=1}^{\frac{M}{2}-1} MS_{\max}^2 (\frac{3T_{\max}^N}{2} + (k-1)\frac{T_{\max}^N}{2})\right)}{\binom{N}{2}} \right.$$
$$+ \frac{\frac{M}{2}S_{\max}^2(\frac{3T_{\max}^N}{2} + (\frac{M}{2}-1)\frac{T_{\max}^N}{2})}{\binom{N}{2}} + \frac{M\binom{S_{\max}}{2}T_{\max}^N}{\binom{N}{2}},$$
$$\left. \frac{3T_{\max}^N}{2} + \left(\frac{M}{2}-1\right)\frac{T_{\max}^N}{2} \right\}. \tag{48}$$

Proof: Consider a configuration of the NFR scheme and any arbitrary Hamiltonian cycle that includes all N nodes. From Eq. (35) we have

$$\begin{aligned} Y_i^{Fu} &= \frac{T}{W_2} \sum_{k=0}^{\frac{M}{2}-1} \sum_{l\in\mathscr{N}(\gamma(i,-k))} \sum_{k'=1}^{\frac{M}{2}-k} \sum_{l'\in\mathscr{N}(\gamma(i,k'))} \lambda(l,l') \\ &\leq \frac{T}{W_2} \sum_{k=0}^{\frac{M}{2}-1} \sum_{l\in\mathscr{N}(\gamma(i,-k))} \lambda \\ &\leq \frac{T}{W_2}\left(\frac{M}{2}\right) S_{\max}\lambda. \end{aligned}$$

On the other hand, Eq. (35) also yields

$$\begin{aligned} Y_i^{Fu} &\leq \frac{T}{W_2} \sum_{k'=1}^{\frac{M}{2}} \sum_{l'\in\mathscr{N}(\gamma(i,k'))} \sum_{k=0}^{\frac{M}{2}-1} \sum_{l\in\mathscr{N}(\gamma(i,-k))} \lambda(l,l') \\ &\leq \frac{T}{W_2} \sum_{k'=1}^{\frac{M}{2}} \sum_{l'\in\mathscr{N}(\gamma(i,k'))} \lambda' \\ &\leq \frac{T}{W_2}\left(\frac{M}{2}\right) S_{\max}\lambda'. \end{aligned}$$

Hence,

$$\frac{Y_i^{Fu}}{T} \leq \frac{(\frac{M}{2})S_{\max}\min(\lambda,\lambda')}{W_2}.$$

Similarly, we can show

$$\begin{aligned} \frac{Y_i^{Bu}}{T} &\leq \frac{(\frac{M}{2})S_{\max}\min(\lambda,\lambda')}{W_2}, \\ \frac{Y_i^{Fd}}{T} &\leq \frac{(\frac{M}{2})S_{\max}\min(\lambda,\lambda')}{W_2}, \\ \frac{Y_i^{Bd}}{T} &\leq \frac{(\frac{M}{2})S_{\max}\min(\lambda,\lambda')}{W_2}. \end{aligned}$$

Therefore,

$$\begin{aligned} \max\left(\frac{Y_i^{Fu}}{T}, \frac{Y_i^{Bu}}{T}\right) &\leq \frac{(\frac{M}{2})S_{\max}\min(\lambda,\lambda')}{W_2}, \\ \max\left(\frac{Y_i^{Fd}}{T}, \frac{Y_i^{Bd}}{T}\right) &\leq \frac{(\frac{M}{2})S_{\max}\min(\lambda,\lambda')}{W_2}. \end{aligned}$$

Using Eqs. (38) and (39), we can write

$$\begin{aligned} X_i^{Fd} + \frac{X_i^{Fl}}{2} &= \frac{T}{W_1} \sum_{l\in\mathscr{N}(i)} \sum_{k=1}^{\frac{M}{2}} \sum_{l'\in\mathscr{N}(\gamma(i,k))} \lambda(l,l') \\ &\quad + \frac{T}{W_1} \sum_{l\in\mathscr{N}(i)} \sum_{l'\in\mathscr{N}_{F,l}(i)} \lambda(l,l') \\ &\leq \frac{T}{W_1} \sum_{l\in\mathscr{N}(i)} \lambda. \end{aligned}$$

Now, from Eqs. (37) and (39) it follows

$$\begin{aligned} X_i^{Fu} + \frac{X_i^{Fl}}{2} &= \frac{T}{W_1} \sum_{l'\in\mathscr{N}(i)} \sum_{k=1}^{\frac{M}{2}} \sum_{l\in\mathscr{N}(\gamma(i,-k))} \lambda(l,l') \\ &\quad + \frac{T}{W_1} \sum_{l'\in\mathscr{N}(i)} \sum_{l\in\mathscr{N}(i)} \lambda(l,l') \\ &\leq \frac{T}{W_1} \sum_{l'\in\mathscr{N}(i)} \lambda'. \end{aligned}$$

Hence,

$$\frac{X_i^F}{T} \leq \frac{S_{\max}(\lambda+\lambda')}{W_1}.$$

Similarly, we can show

$$\frac{X_i^B}{T} \leq \frac{S_{\max}(\lambda+\lambda')}{W_1}.$$

Therefore,

$$\max\left(\frac{X_i^F}{T}, \frac{X_i^B}{T}\right) \leq \frac{S_{\max}(\lambda+\lambda')}{W_1}.$$

Finally, using Eqs. (32), (33), (46), and the inequalities above, we obtain

$$T \leq \frac{\frac{L}{V}}{\frac{1}{2} - S_{\max}\left[\frac{M\min(\lambda,\lambda')}{W_2} + \frac{\lambda+\lambda'}{W_1}\right]} \triangleq T_{\max}^N.$$

A packet needs to be relayed by at most $M/2$ MFs before reaching its destination, so the maximum number of MFs that carry this packet is $(\frac{M}{2}+1)$ (considering also the MF that collects it from the source node). Each MF carries the packet for at most $T/2$ time units, and it waits for at most T time units before it is collected. Thus,

$$D_{\max} \leq \left(\frac{M}{2}+1\right)\frac{T}{2}+T \leq \left(\frac{M}{4}+\frac{3}{2}\right)T^N_{\max}.$$

The average waiting time of a packet before it is picked up is $T/2$ and if its destination is within the origin path, then the maximum time to deliver the packet after it is picked up is less than $T/2$. However, if its destination is k paths $(k = 1,\ldots,\frac{M}{2})$ away from the origin path, then the average delay is less than $\frac{3T}{2}+(k-1)(\frac{T}{2})$. The former case includes at most $M\binom{S_{\max}}{2}$ SD pairs. The latter case includes at most $MS^2_{\max}$ SD pairs for each $k = 1,\ldots,(\frac{M}{2}-1)$, and at most $\frac{M}{2}S^2_{\max}$ SD pairs for $k = \frac{M}{2}$. Also, there are a total of $\binom{N}{2}$ SD pairs in the network. Thus,

$$\mathbf{E}[D] \leq \frac{\left(\sum_{k=1}^{\frac{M}{2}-1} MS^2_{\max}(\frac{3T^N_{\max}}{2}+(k-1)\frac{T^N_{\max}}{2})\right)}{\binom{N}{2}} + \frac{\frac{M}{2}S^2_{\max}(\frac{3T^N_{\max}}{2}+(\frac{M}{2}-1)\frac{T^N_{\max}}{2})}{\binom{N}{2}} + \frac{M\binom{S_{\max}}{2}T^N_{\max}}{\binom{N}{2}}. \tag{49}$$

Taking into account (49), considering that the average delay does not exceed $\frac{3T^N_{\max}}{2}+(\frac{M}{2}-1)(T^N_{\max}/2)$ no matter where the destination is, and noting that the total number of SD pairs is $\binom{N}{2}$, we obtain (48). ■

A result analogous to Corollary IV.2 can be obtained for Thm. VI.1 as well.

Corollary VI.2. *Suppose that the data rates between the nodes are random variables and the maximum output rate from and maximum input rate to any arbitrary node, denoted by $\tilde{\lambda}$ and $\tilde{\lambda}'$ (cf. (7)), respectively, satisfy*

$$\tilde{\lambda} \leq \lambda, \quad \tilde{\lambda}' \leq \lambda' \qquad \text{w.p.1.}$$

Assume further that the maximum length of an MF path, $\tilde{L}$, is also stochastic such that $\tilde{L} \leq L$, w.p.1. Then, bound (47) on the worst-case delay holds w.p.1.

Furthermore, using Thm. IV.4 we can show the following.

Corollary VI.3. *Assume N nodes are placed arbitrarily on a unit square, each of which transmitting with a rate of λ to other nodes (such that the data rate between any pair of nodes is the same). If $\lambda < \frac{1}{2S_{\max}(\frac{M}{W_2}+\frac{2}{W_1})}$, then there exists an NFR scheme for which (47) and (48) hold with $T^N_{\max}$ replaced by*

$$\tilde{T}^N_{\max} = \frac{\frac{\sqrt{2S_{\max}}+\frac{7}{4}}{V}}{\frac{1}{2}-S_{\max}\left[\frac{M\lambda}{W_2}+\frac{2\lambda}{W_1}\right]}.$$

B. NFR Optimization

NFR optimization is defined as the problem of finding a configuration of the NFR scheme (finding a Hamiltonian cycle through the nodes and splitting it into appropriate paths) such that T in Eq. (46) is minimized. As the earlier schemes, NFR optimization is NP-hard (reduction from TSP) and we again rely on a heuristic algorithm. Intuitively, good paths should be short and constructed such that nodes with high data flow among themselves are located closer to each other. The former helps us reduce the MFs' travel times, and the latter reduces inter-MF relay times. Our heuristic algorithm consists of two phases. In the first phase, we find a good Hamiltonian cycle and in the second phase we assign the nodes to the MFs.

1) Phase I (Choosing a good Hamiltonian cycle): We first use well known TSP heuristics [37], [38] which compute a cycle visiting all nodes. However, instead of optimizing the length of the cycle route as in TSP, we optimize the following estimate (τ) of the cycle time:

$$\tau = \frac{\frac{\tilde{L}}{V}}{\frac{M}{2}-[\frac{\sum_{i=1}^N\sum_{j=1}^N y(i,j)\lambda(i,j)}{W_2}+\frac{\sum_{i=1}^N\sum_{j=1}^N \lambda(i,j)}{W_1}]},$$

which is obtained by solving the following equation for τ:

$$\tau = \frac{2}{M}\frac{\tilde{L}}{V}+\frac{1}{(\frac{M}{2})}\left(\tau\frac{\sum_{i=1}^N\sum_{j=1}^N y(i,j)\lambda(i,j)}{W_2}\right) + \frac{2}{M}\left(\tau\frac{\sum_{i=1}^N\sum_{j=1}^N \lambda(i,j)}{W_1}\right). \tag{50}$$

In the equation above, $\tilde{L}$ is the length of the Hamiltonian cycle, and $y(i,j)$ returns an estimate of the average number of relays needed to deliver data from node i to node j along the cycle, defined as

$$y(i,j) = \frac{s(i,j)}{N/M},$$

where $s(i,j)$ is the number of nodes located between node i and node j along the direction of the cycle that yields the smaller number. The estimate in Eq. (50) can be intuitively understood as follows. Assuming that we split the cycle into paths of equal length, each MF's travel times, A^F_i and A^B_i, would be equal to $\frac{\tilde{L}}{VM}$, and each MF's total travel time (forward and backward) in one cycle would be equal to $\frac{2\tilde{L}}{VM}$. Also, in one cycle, there would be $\tau\sum_{i=1}^N\sum_{j=1}^N y(i,j)\lambda(i,j)$ bits of data to be relayed among the M MFs, and since in the inter-MF relay process two MFs are communicating with each other, each MF would spend $\frac{1}{(\frac{M}{2})}(\tau\frac{\sum_{i=1}^N\sum_{j=1}^N y(i,j)\lambda(i,j)}{W_2})$ units of time as its inter-MF relay time. Note that as $s(i,j)$ grows, $y(i,j)$, the average number of relays from node i to node j, also grows. In addition, in one cycle, there would be a total of $\tau\sum_{i=1}^N\sum_{j=1}^N\lambda(i,j)$ bits of data to be delivered to all nodes by M MFs, and since the data should be also collected in order to be delivered, each MF would spend $\frac{2}{M}(\tau\frac{\sum_{i=1}^N\sum_{j=1}^N\lambda(i,j)}{W_1})$ units of time in each cycle as its MF-node transfer time.

Similar to [15], we consider the following 2-opt and 2H-opt swap operations to improve the cycle.

- *2-opt swap*: A 2-opt swap removes two edges AB and CD from the cycle and replaces them with edges AC and BD while maintaining a single cycle.
- *2H-opt swap*: A 2H-opt swap moves a node in the path from one position to another.

Thus, the algorithm tries to reduce τ by applying 2-opt and 2H-opt swaps until no further improvement is found. There are no complexity results for the TSP heuristics we discussed

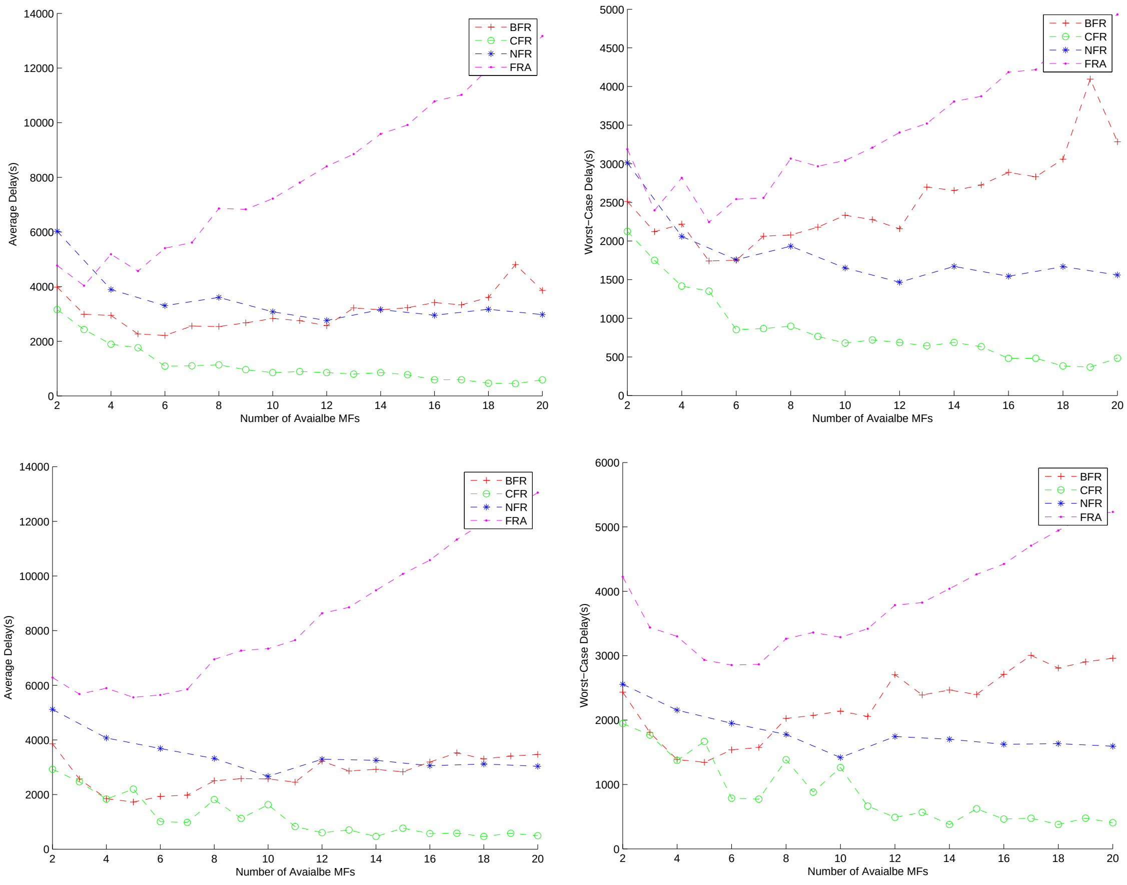


Fig. 4. Comparison of the schemes for the scenario with topology U and the uniform data model: (Upper Left) Average delay; (Upper Right) Worst-case delay; and of the schemes for the scenario with topology A and the uniform data model: (Lower Left) Average delay; (Lower Right) Worst-case delay.

but one can limit the number of iterations to maintain low complexity.

2) Phase II (Assigning nodes to MFs): This is essentially a local search heuristic. We start from an arbitrary point in the cycle obtained in Phase I and as we move in an arbitrary direction we assign the first $\frac{N}{M}$ nodes to MF 1, the next $\frac{N}{M}$ nodes to MF 2, and so on and so forth. We assume that two MFs meet with each other at a contact point defined as the midpoint of the edge that is incident to both of their corresponding paths. Given this construction, each MF's cycle time T_i can be calculated using Eq. (34), and the actual total cycle time T is obtained from Eq. (46). Then, the MF with the largest T_i is recognized. We call this MF the "critical MF" and its path the "critical path." We target the critical path and try to shorten the critical MF's cycle time by de-assigning one of its two "extreme" nodes (the nodes that are located closest to the MF's forward and backward neighbors) and assigning them to the corresponding (closest) neighbor. If this node-exchange reduces the total cycle time, then we apply the exchange; otherwise we attempt this node-exchange process between neighboring paths that have the largest difference between their cycle times. In the latter case, such a node-exchange is allowed only if it does not increase the total cycle time and results in a smaller difference between the cycle times of these paths. We continue this process until no further improvement is realized. Note that the proposed approach always achieves the average and worst-case delays given in Theorem VI.1. As we have done so far, we can limit the number of iterations of this assignment phase in order to maintain low complexity.

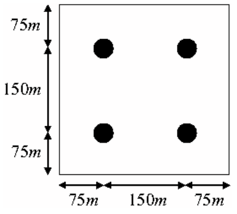


Fig. 6. The locations of the cluster centers in topology A.

VII. Numerical Evaluation

In this section, we evaluate the effectiveness of the schemes we introduced via simulation under various scenarios. The simulation was developed in the MATLAB environment. A graph generator forms the random network topology and determines the data rates according to a given scenario. Given node positions and pairwise data rates, the CFR, BFR, and NFR optimization problems are solved using the proposed algorithms. We consider 40 nodes distributed on a $300m \times 300m$ area. Two topologies, A and U, are assumed in our simulations.

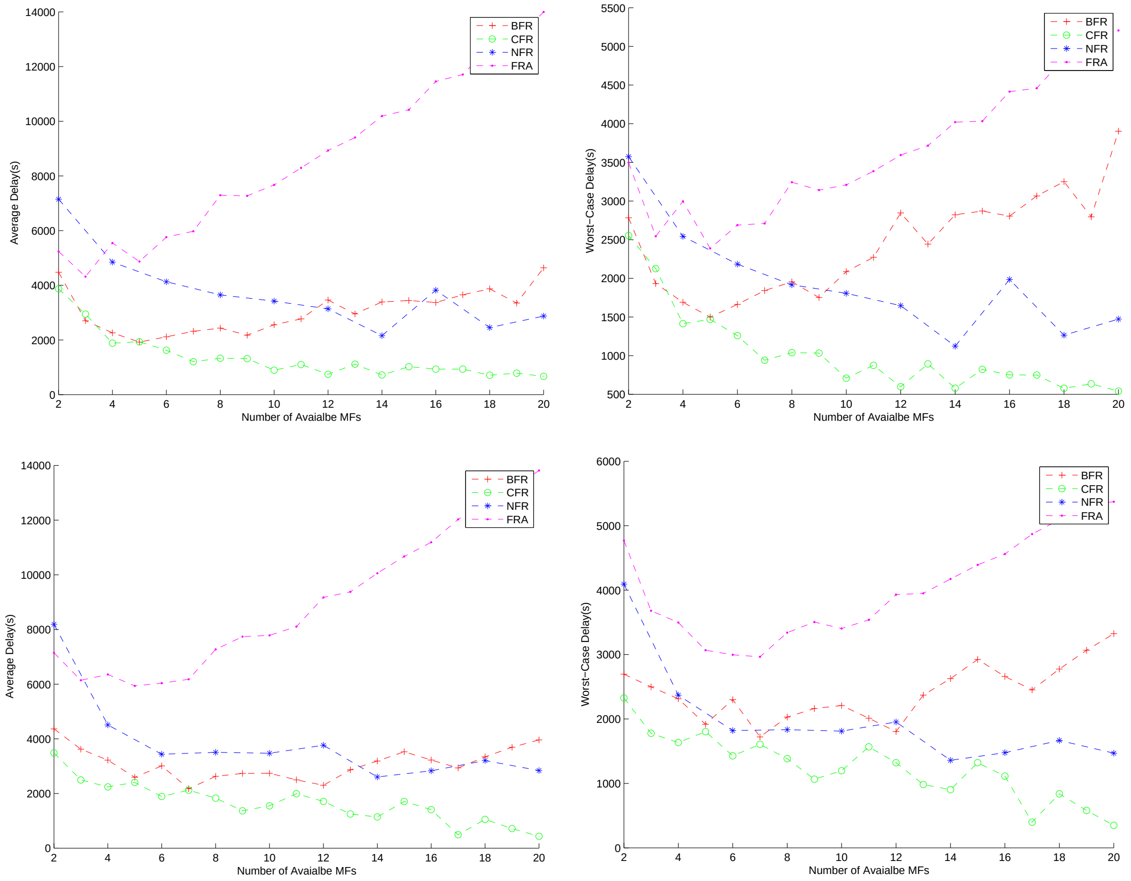


Fig. 5. Comparison of the schemes for the scenario with topology U and the non-uniform data model: (Upper Left) Average delay; (Upper Right) Worst-case delay; and of the schemes for the scenario with topology A and the non-uniform data model: (Lower Left) Average delay; (Lower Right) Worst-case delay.

Topology U corresponds to uniformly distributed nodes over the coverage area. In Topology A, we assume that the nodes are concentrated at certain locations called *cluster centers*. The cluster centers have the highest density of nodes, which drops radially outward. More specifically, the area was divided into four $150m \times 150m$ squares, and the centers of the four squares were chosen as the locations of the cluster centers (see Fig. 6). To generate topologies A, x and y coordinates of each of the 10 nodes belonging within each cluster are generated from a Normal distribution with mean equal to the corresponding coordinate of their cluster center and standard deviation equal to 20m. The communication between MFs and nodes is done at the rate of $W_1 = 20$ *Mbps* and among MFs at the rate of $W_2 = 20$ *Mbps* also. We consider two data traffic models in the network: *uniform flow* and *non-uniform flow*. In the uniform flow model, the data rate between any two nodes is randomly selected from a Uniform distribution that takes values between 0 and 1 *Kbps*, whereas in the non-uniform flow model, 10% of the traffic has a rate of 3 *Kbps* and the rest a rate of 1 *Kbps*. We set $V = 1(m/s)$. The performance measures we computed via simulation include the average delay and the worst-case delay.

Figs. 4 and 5 depict the average and worst-case delay for different scenarios as the number of MFs increases. As we noted earlier, for the NFR scheme we require M to be even so that at each contact point two adjacent MFs are available to relay data to each other. The results show that the CFR, BFR, and NFR schemes generally yield substantially less average delay than the FRA scheme. This gain is generally more pronounced for larger values of M. It appears that the FRA scheme is the least capable of utilizing the additional number of MFs to decrease the delay (when the number of MFs becomes large, the delay increases in the FRA scheme); this is due to excessive inter-MF relays in the FRA scheme. Notice that in the BFR scheme the average delay also increases after a certain number of MFs, because we utilize a single bus which has to travel between many MFs. We discussed earlier (see Sec. V) how this issue can be addressed by employing multiple buses.

Regarding worst-case delay, the qualitative conclusions are similar. Note that CFR and NFR keep the worst-case delay low while also maintaining a low average delay. BFR (with a single bus) increases worst-case delay after a number of MFs get employed, but still outperforms FRA. BFR may be appropriate for a specific application, especially when vehicles that roam the environment (e.g., real buses) already exist. These observations guide the network designer to select an appropriate scheme given performance requirements and application considerations.

Fig. 7 shows the ratio of simulated worst-case delay and

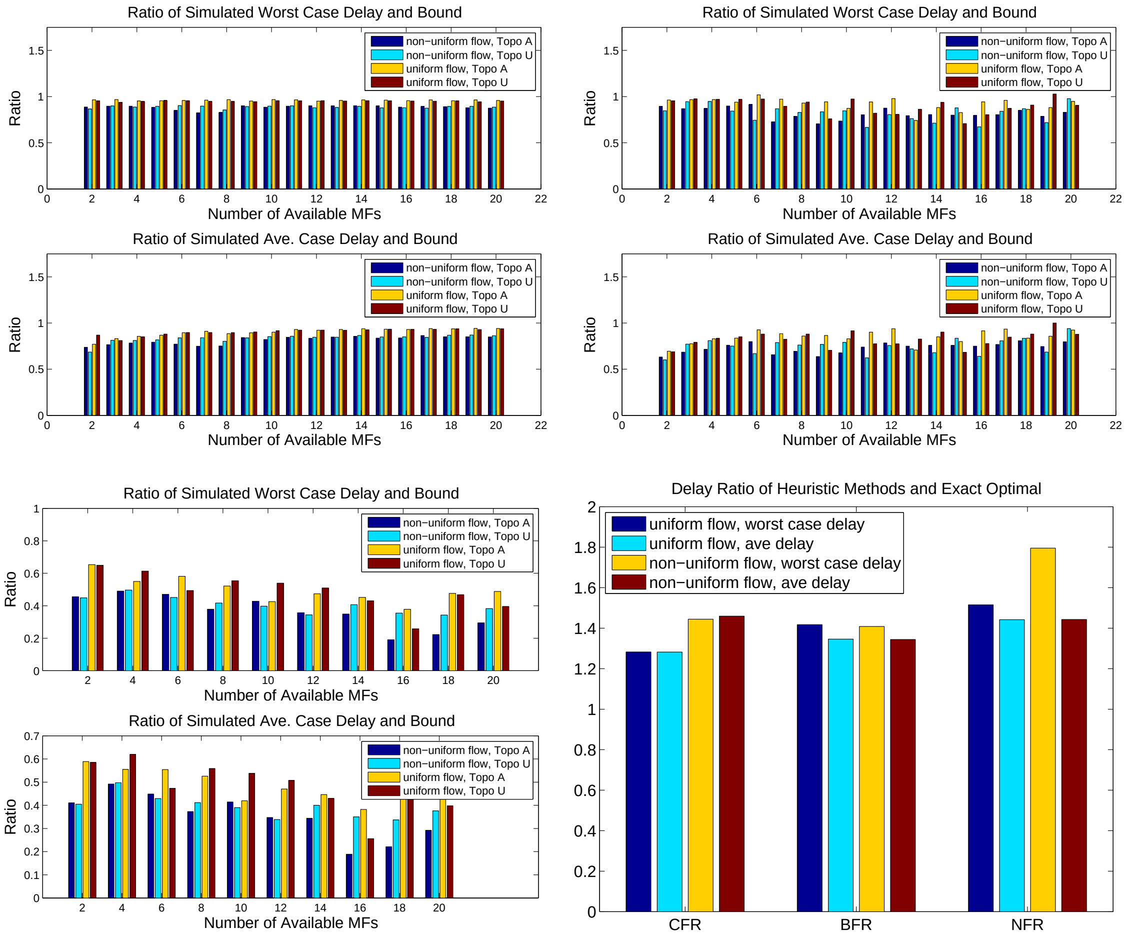


Fig. 7. Assessing the tightness of the bounds: (Upper Left) CFR; (Upper Right) BFR; (Lower Left) NFR. In each plot and for each number of available MFs the 1st and 2nd bars correspond to non-uniform flow and Topology A and U, respectively, and the 3rd and 4th bars correspond to uniform flow and Topology A and U, respectively. (Lower Right): Assessing the effectiveness of CFR, BFR, and NFR optimization approaches. The 1st and 2nd bars correspond to uniform flow and worst-case and average delay, respectively, and the 3rd and 4th bars to non-uniform flow for the two respective delay metrics.

average delay over the corresponding upper bounds we have established for CFR, BFR, and NFR, respectively (cf. Theorems IV.1, V.1, and VI.1, respectively). In general, the bounds under uniform flow traffic are tighter than the ones under non-uniform flows, which is to be expected based on our earlier discussion. The bounds of CFR are the tightest and the gap is less than 30%. For BFR and NFR, the bounds are tight in most cases.

Finally, we assess the effectiveness of the CFR, BFR, and NFR optimization approaches we have proposed in Sections IV-C, V-B, and VI-B, respectively. In each case we derived an optimal policy using a brute-force method for a small-scale network with 4 nodes and 2 MFs. In the brute-force method, we calculate the total cycle time for all possible configurations and select the configuration with the smallest such value. This calculation of total cycle time is done by simulation and this introduces some (estimation) error which may prevent us from obtaining the true optimal policy. Yet, the policy we derive should be very close to optimal and provides an appropriate baseline. In this comparison, the network topology is set to be Topology U and both uniform and non-uniform flow models are considered. The results appear in Fig. 7 (Lower Right), where it can be seen that our heuristic methods produce policies that are not that far from optimal (by a factor of 1–1.5 for CFR and BFR and a factor of 1.4–1.8 for NFR).

VIII. Conclusions

In this paper, we studied the problem of designing routes for mobile elements (message ferries) in wireless networks. We considered a general network model where each node in the network sends data to multiple other nodes with possibly different data rates. We proposed schemes that utilize useful interactions among the ferries. For each scheme, analytical bounds on worst-case and average delay were derived. Worst-case bounds in particular, have not received enough attention in the literature even though they are critical in offering performance guarantees.

We also established that one of our schemes (CFR) has desirable scalability properties; achieving constant per-node throughput within constant maximum (worst-case) delay as the network size grows. Simulation results for several qualitatively different topologies and traffic patterns showed that our schemes perform better than the alternative Ferry Relaying Algorithm proposed earlier in the literature.

We noted that the BFR scheme can be possibly improved if more than one bus is used when the number of ferries is relatively large (e.g., using two buses each of which traveling half of the route designed for the single bus case). In addition, and for all of our schemes, it would also be interesting to explore route design where some of the nodes (e.g., the ones with higher data rates) are visited more than once in every cycle. These are interesting directions to be pursued in future research.

Acknowledgments

We would like to thank the anonymous reviewers whose questions and comments helped us improve our original results.

References

[1] J. Hart and K. Martinez, "Environmental Sensor Networks: A revolution in the earth system science?" *Earth-Science Reviews*, vol. 78, no. 3-4, pp. 177–191, 2006.

[2] M. Dunbabin, P. Corke, I. Vasilescu, and D. Rus, "Data muling over underwater wireless sensor networks using an autonomous underwater vehicle," in *Proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA)*, 2006, pp. 2091–2098.

[3] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with Zebranet," in *International Conference on Architectural support for programming languages and operating systems*, 2002.

[4] M. Baard, "Motes make trash cans smarter," Boston Globe, June 5 2006.

[5] 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, no. 2, pp. 215–233, 2003.

[6] W. Zhao and M. Ammar, "Message ferrying: proactive routing in highly-partitioned wireless ad hoc networks," in *Proceedings of the 9th IEEE workshop on future trends of distributed computing systems*, 2003, pp. 308–314.

[7] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in *Proceedings of the 5th ACM international symposium on mobile ad hoc networking and computing*, 2004, pp. 187–198.

[8] I. Akyildiz and I. Kasimoglu, "Wireless sensor and actor networks: research challenges," *Ad Hoc Networks Journal*, vol. 2, pp. 351–367, 2004.

[9] E. Ngai, Y. Zhou, M. Lyu, and J. Liu, "Reliable reporting of delay-sensitive events in wireless sensor-actuator networks," in *Proceedings of the 3rd IEEE MASS*, 2006.

[10] Y. Shi and Y. T. Hou, "Theoretical results on base station movement problem for sensor network," in *Proceedings of the IEEE Infocom*, Phoenix, AZ, April 14-17 2008, pp. 376–384.

[11] W. Saad, Z. Han, T. Basar, M. Debbah, and A. Hjerungnes, "Hedonic coalition formation for distributed task allocation among wireless agents," IEEE Trans. on Mobile Computing, preprint, 2011.

[12] A. Beafour, M. Leopold, and P. Bonnet, "Smart tag based data dissemination," in *ACM Workshop on Wireless Sensor Networks and Applications*, 2002.

[13] "Daknet project," http://web.media.mit.edu/∼amir/daknet/, 2009.

[14] M. Todd, D. Mascarenas, E. Flynn, T. Rosing, B. Lee, D. Musiani, S. Dasgupta, S. Kpotufe, D. Hsu, R. Gupta, G. Park, T. Overly, M. Nothnagel, and C. Farrar, "A different approach to sensor networking for SHM: Remote powering and interrogation with unmanned aerial vehicles," in *Proceedings of the 6th International workshop on Structural Health Monitoring*, 2007.

[15] W. Zhao, M. Ammar, and E. Zegura, "Controlling the mobility of multiple data transport ferries in a delay-tolerant network," in *Proceedings of IEEE INFOCOM 2005*, vol. 2, 2005, pp. 1407–1418.

[16] A. Pandya, A. Kansal, and G. Pottie, "Goodput and delay in networks with controlled mobility," in *2008 IEEE Aerospace Conference*, 2008, pp. 1–8.

[17] Y. Chen, J. Yang, W. Zhao, M. Ammar, and E. Zegura, "Multicasting in sparse MANETs using message ferrying," in *Wireless Communications and Networking Conference*, 2006, pp. 691–696.

[18] S. Parragh, K. Doerner, and R. Hartl, "A survey on pickup and delivery problems: Part I: Transportation between customers and depot," *Journal fur Betriebswirtschaft*, vol. 58, no. 1, pp. 21–51, 2008.

[19] S. Parragh, K. F. Doerner, and R. Hartl, "A survey on pickup and delivery problems: Part II: Transportation between pickup and delivery locations," *Journal fur Betriebswirtschaft*, vol. 58, no. 2, pp. 81–117, 2008.

[20] G. Berbeglia, J. Cordeau, I. Gribkovskaia, and G. Laporte, "Static pickup and delivery problems: a classification scheme and survey," *TOP*, vol. 15, no. 1, pp. 1–31, 2007.

[21] P. Gupta and P. Kumar, "The capacity of wireless networks," *IEEE Trans. Inform. Theory*, vol. 46, pp. 388–404, 2000.

[22] M. Grossglauser and D. Tse, "Mobility increases the capacity of ad-hoc wireless networks," in *IEEE INFOCOM*, 2001, pp. 1360–1369.

[23] A. Gamal, J. Mammen, B. Prabhakar, and D. Shah, "Throughput-delay trade-off in wireless networks," in *IEEE INFOCOM*, 2004.

[24] M. Neely and E. Modiano, "Capacity and delay tradeoffs for ad-hoc mobile networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, 2005.

[25] G. Sharma, R. Mazumdar, and N. Shroff, "Delay and capacity trade-offs in mobile ad hoc networks: A global perspective," in *IEEE INFOCOM*, 2006.

[26] N. Bansal and Z. Liu, "Capacity, delay and mobility in wireless ad-hoc networks," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications*, vol. 2, 2003, pp. 1553–1563.

[27] A. Schrijver, *Combinatorial Optimization*. Springer, 2003.

[28] L. Few, "The shortest path and the shortest road through n points in a region," *Mathematika*, vol. 2, pp. 141–144, 1955.

[29] L. Fejes-Toth, "Uber einen geometrische satz," *Math. Z.*, vol. 46, pp. 83–85, 1940.

[30] M. Steele, "Probabilistic and worst case analyses of classical problems of combinatorial optimization in euclidean space," *Math. Operations Research*, vol. 15, no. 4, pp. 749–770, 1990.

[31] C. Noon and J. Bean, "An efficient transformation of the generalized traveling salesman problem," *INFOR*, vol. 31, no. 1, pp. 39–44, 1993.

[32] Y. Lien, E. Ma, and B. Wah, "Transformation of the generalized traveling salesman problem into the standard traveling salesman problem," *Information Science*, vol. 74, pp. 171–189, 1993.

[33] V. Dimitrijevic and Z. Saric, "An efficient transformation of the generalized traveling salesman problem into the traveling salesman problem on digraphs," *Information Science*, vol. 102, pp. 105–110, 1997.

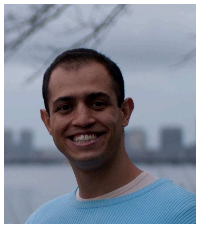
[34] G. Laporte and F. Semet, "Computational evaluation of a transformation procedure for the symmetric generalized traveling salesman problem," *INFOR*, vol. 37, no. 2, pp. 114–120, 1999.

[35] A. Behzad and M. Modarres, "A new efficient transformation of the generalized traveling salesman problem into traveling salesman problem," in *Proceedings of the 15th International Conference of Systems Engineering*, 2002, pp. 6–8.

[36] J. Renaud, F. Boctor, and G. Laporte, "A fast composite heuristic for the symmetric traveling salesman problem," *INFORMS Journal on Computing*, vol. 8, no. 2, pp. 134–143, 1996.

[37] J. Bentley, "Fast algorithms for geometric traveling salesman problems," *INFORMS Journal on Computing*, vol. 4, no. 4, pp. 387–411, 1992.

[38] D. Johnson and L. McGeoch, *Experimental analysis of heuristics for the STSP, The traveling salesman problem and its variations*. Springer US, 2002.



Reza Moazzez-Estanjini received the B.Sc. degree in Industrial Engineering from Sharif University, Tehran, Iran, in 2007 and the Ph.D. degree from the Division of Systems Engineering, Boston University, Boston, MA, in 2011. His research interests include control and optimization with applications to wireless networks and vehicle routing. He received the Student Paper Award at the 2011 IEEE WiOpt Conference.

Jing (Conan) Wang is a Ph.D. Student in the Division of Systems Engineering at Boston University. He received his B.E. in Electrical and Information Engineering from Huazhong University of Science and Technology, China, in 2006. His research interests include cyber security and approximate dynamic programming.

Ioannis Ch. Paschalidis (M'96, SM'06) is a Professor and Distinguished Faculty Fellow at Boston University with appointments in the Department of Electrical and Computer Engineering and the Division of Systems Engineering. He is a Co-Director of the Center for Information and Systems Engineering (CISE). He completed his graduate education at the Massachusetts Institute of Technology (MIT) receiving an M.S. (1993) and a Ph.D. (1996), both in Electrical Engineering and Computer Science. In September 1996 he joined Boston University where he has been ever since. He has held visiting appointments with MIT, and the Columbia University Business School. His current research interests lie in the fields of systems and control, networking, applied probability, optimization, operations research and computational biology.