Dispatch-and-Search: Dynamic Multi-Ferry Control in Partitioned Mobile Networks

Ting He†, Ananthram Swami†, and Kang-Won Lee†
† IBM Research, Hawthorne, NY 10532 USA
‡ Army Research Laboratory, Adelphi, MD 20783 USA
{the,kangwon}@us.ibm.com
aswami@arl.army.mil

ABSTRACT

We consider the problem of disseminating data from a base station to a sparse, partitioned mobile network by controllable data ferries with limited ferry-node and ferry-ferry communication ranges. Existing solutions to data ferry control mostly assume the nodes to be stationary, which reduces the problem to designing fixed ferry routes. In the more challenging scenario of mobile networks, existing solutions have focused on single-ferry control and left out an important issue of ferry cooperation in the presence of multiple ferries. In this paper, we jointly address the issues of ferry navigation and cooperation using the approach of stochastic control. Under the assumption that ferries can communicate within each partition, we propose a hierarchical control system called Dispatch-and-Search (DAS), consisting of a global controller that dispatches ferries to individual partitions and local controllers that coordinate the search for nodes within each partition. Formulating the global and the local control as Partially Observable Markov Decision Processes (POMDPs), we develop efficient control policies to optimize the (discounted) total throughput, which significantly improve the performance of their predetermined counterparts in cases of limited prior knowledge.

Categories and Subject Descriptors
C.2.1 [Computer-Communication Networks]: Network Architecture and Design—store and forward networks; G.4 [Mathematical Software]: algorithm design and analysis; I.2.8 [Artificial Intelligence]: Problem Solving, Control Methods, and Search—control theory; I.2.9 [Artificial Intelligence]: Robotics—autonomous vehicles

General Terms
Algorithms, Design, Performance

Keywords
Data ferry control, Partially Observable Markov Decision Processes, Myopic control policies

1. INTRODUCTION

The development of robotic technology has led to improvements in many applications, a recent one being the use of controllable mobile nodes to assist communications in sparse networks. Sparse network topologies are routinely encountered in a broad variety of applications, e.g., sensor/actuator networks, search-and-rescue operations, underwater networks, spacecraft networks ("interplanetary communications"), vehicular ad hoc networks (VANETs), and military mobile ad hoc networks (MANETs). The sparse networks discussed above pose significant challenges to traditional communication schemes since existing in-network solutions such as routing and delay-tolerant communications are only designed for (intermittently) connected networks and will fail if the network is permanently partitioned. In such partitioned networks, the only remedy is to look for external help, and mobile helper nodes called data ferries, usually mounted on controllable platforms such as UAVs, stand out as a preferred solution in many scenarios due to their agility in contrast to fixed infrastructures. The problem, though, is that proper control has to be in place to leverage these data ferries effectively.

Previous solutions to data ferry control have mostly focused on designing fixed ferry routes [3, 5, 12, 13] for static networks. Although the approach still applies to mobile networks [9], the performance is generally suboptimal as fixed routes can only leverage long-term statistics of node movements. To fully utilize the ferries' capability, we have shown in [2] that each ferry needs to dynamically adjust its movements based on run-time observations. As in [2], we assume a ferry can only observe nodes within a certain distance defined by ferry-node communication range, referred to as partial observations¹. Although it shows promising performance improvements over fixed routes, the solution in [2] is limited by (i) only considering a single ferry and (ii) only focusing on contact rate optimization. The availability of multiple ferries raises a new challenge because one

¹This is in contrast to assuming complete observations of nodes in the entire network as in [1, 11, 14].
ferry’s optimal movements will depend on the movements of the other ferries, which induces a need for the ferries to cooperate. While this is not an issue under unlimited ferry-ferry communications, global cooperation can be expensive, especially for networks with scattered partitions. Furthermore, although contact rates are performance indicators, the ultimate goal of the ferries should be to optimize the end-to-end communication performance perceived by nodes, which also depends on other factors such as traffic demands. These new challenges call for a new solution that can efficiently utilize multiple ferries to optimize the end-to-end performance without excessive cooperation overhead.

1.1 Related Work

Designated communication nodes, typically called data ferries or data mules, have been considered as an effective solution to support communications in sparse networks. Data ferries physically carry data either between task (i.e., non-ferry) nodes [3, 11–14] or between nodes and a base station [1, 4–6]. The main issue in using such ferries is how to control their mobility, where existing solutions can be divided into single-ferry control [1, 2, 4, 9, 14] and multi-ferry control [3, 5, 6, 12, 13]. An in-between case was studied in [11] where nodes can switch between the roles of ferry and non-ferry. Most existing work focuses on ferry route design under the assumption that either the task nodes are stationary [3–5, 12, 13], or their locations are always known by the ferries if they are mobile (i.e., complete observations) [11, 12]. A related problem of static but randomly distributed nodes has been studied: using predetermined routes in [6] and dynamic route adaptation based on complete observations in [1]2. The problem of stochastic node movements and partial observability was first studied in [9] using predetermined routes and later improved in [2] using a control policy that navigates the ferry dynamically.

1.2 Our Approach and Results

We consider controlling multiple ferries in partitioned mobile networks to optimize the total throughput. We present our solution in the scenario of data dissemination, although our approach extends naturally to the cases of data harvesting and peer-to-peer communications (see discussions in Section 7). Our specific contributions are:

Flexible control framework: To avoid long-range ferry-ferry communications, we propose a hierarchical control framework called Dispatch-and-Search (DAS), which divides the problem into the global control of allocating ferries among partitions and the local control of searching for nodes within each partition. Since the global control can be implemented by the base station (BS), ferries only need to communicate within each partition to cooperate in local control.

Rigorous problem formulation: Under Markovian node mobility, we show that both the global and the local control problems can be formulated as special cases of Partially Observable Markov Decision Process (POMDP). The POMDP model provides a comprehensive representation of available information, control options, and performance objective in a dynamic system, which provides a foundation for rigorous optimization.

Efficient solutions: Based on the POMDP formulation,

2Both works treat each message as a new node which arrives randomly in space and time; [1] also assumes locations and arrival times of pending messages are known at run time.

2. PROBLEM FORMULATION

2.1 Network Model

Suppose that there is a total of $K$ ferries and $D$ network partitions, each partition occupying a disjoint region called a domain, as illustrated in Fig. 1. A base station (BS) generates traffic for all nodes at constant rates $\lambda = (\lambda_d)_{d=1}^{D}$ (fluid model), which is then delivered to the respective domains by the ferries in a delay-tolerant manner. In each domain, we select a node as the gateway to the ferries, which then disseminates received traffic within the domain using in-network routing; we will simply call these gateway nodes “nodes”. According to the ferry-node communication range, each domain is partitioned into $N_d$ ($d = 1, \ldots, D$) cells denoted by a set $S_d$ $(S_d \cap S_{d'} = \emptyset \text{ for } d \neq d')$, such that the transmission of each ferry can cover one cell at a time. We

\[ r_i = \text{ (projected) ferry communication radius).} \]
assume that node mobility in domain \( d \) can be modeled as a Markov chain on \( S_d \) with transition matrix \( P_d \). We also assume that ferry-ferry communications can span an entire domain but not across domains. The total network field is denoted by \( \mathcal{S} := \bigcup_{d=1}^{D} S_d \) of size \( |\mathcal{S}| = N = \sum_{d=1}^{D} N_d \). We do not restrict the relationships between \( K, D, \) and \( N_d \), although restrictions may apply in later analysis (e.g., Section 5.3).

2.2 System Architecture

Due to the ferry-ferry communication constraint, ferries can only cooperate locally within a domain. Accordingly, we propose a hierarchical control system called “Dispatch-and-Search” (DAS), consisting of a global dispatch controller \( \pi_d \) at the BS in charge of allocating ferries to individual domains, and a local search controller \( \pi_{d,g} \) for each domain \( d \), located at an arbitrarily selected lead ferry, in charge of jointly navigating ferries dispatched to that domain. Specifically, the lead ferry will collect observations from the other ferries in its domain and broadcast navigation actions determined by the search controller in terms of the set of cells to cover (and their matching with the ferries) in each slot in order to contact the gateway node (by any ferry) and deliver traffic. We assume that all ferries dispatched to the same domain carry the same traffic. As illustrated in Fig. 2, dispatch control operates periodically with period \( \Delta + 1 \) slots, called a round (\( \Delta \) is a design parameter); the first \( \Delta \) slots are used to search for the gateway nodes in the individual domains; the last slot is used for all the ferries to return to the BS and pick up new traffic, after which the next round begins.

2.3 Main Problem: DAS Control

Let \( \lambda_\beta := \mathbb{E} [\sum_{t=1}^{\infty} N^* (t) \beta^t] \) denote the discounted total throughput for a given discount factor \( \beta \in (0, 1) \) under a DAS control policy \( \pi := (\pi_d, (\pi_{d,g})_{d=1}^{D}) \), where \( N^*(t) \) is the amount of data delivered (over all domains) at time \( t \) under policy \( \pi \). Our goal is to design a policy \( \pi \) that maximizes \( \lambda_\beta \).

Remarks: The discounts are used to model the objective of delivering the data as soon as possible. The above definition is a cumulative throughput, but it represents a delivery rate, defined as the ratio of the delivered traffic and the delivery time \( \lambda_\beta := \mathbb{E} [(\sum_{t=1}^{\infty} N^* (t) \beta^t)/(\sum_{t=1}^{\infty} \beta^t)] \), in that \( \lambda_\beta = \lambda_\beta/(1 - \beta) \).

3. PRELIMINARY

3.1 POMDP Recap

POMDP [8] is a general framework to model the control of a stochastic system. We provide a quick overview of POMDP in this section. A POMDP is represented by a tuple \((\mathcal{X}, \mathcal{A}, \mathcal{O}, r, P_a, P_o)\), where \( \mathcal{X} \) represents the possible states of the controlled system, \( \mathcal{A} \) the set of control actions, \( \mathcal{O} \) the observations of controller, and \( r(x, a, o) \) a reward function modeling the performance criterion based on the state \( x \in \mathcal{X} \), the action \( a \in \mathcal{A} \), and the observation \( o \in \mathcal{O} \).

The objective of the controller is to find a policy \( \pi \) that determines the action to take based on the current belief so as to maximize the total reward over a time window. With an infinite horizon, a common form of the total reward is the discounted reward

\[
R_\infty = \mathbb{E} \left[ \sum_{t=1}^{\infty} \beta^t r(x_t, a_t^\pi, o_t) \right]
\]

for a fixed discount factor \( \beta \in (0, 1) \). The optimal policy \( \pi^\star \) that maximizes \( R_\infty \) is known to be the solution to the Bellman equation:

\[
V_\infty (b) = \beta \arg \max_a \mathbb{E} [r(b, a, o) + V_\infty (T(b, a, o))],
\]

where \( V_\infty (b) := \max_a R_\infty (b, a) \), called the value function, is the maximum reward starting from a given belief state \( b \), the expectation is over \( P_o(b, a, o) := \sum_{x} b(x) P_o(x, a, o) \), and the function \( b' = T(b, a, o) \) generates a new belief for the next step based on the previous belief, the action, and the observation. It is known that solving (2) for the optimal policy is PSPACE-hard in general [7]. A popular alternative is to use the myopic policy in which one only maximizes the average immediate reward, and is given by

\[
\pi_{MY} (b) := \arg \max_a \sum_{x} b(x) \mathbb{E} [r(x, a, o)],
\]

where the expectation is over \( P_o(x, a, o) \). The myopic policy is easy to implement and has exhibited excellent performance in single-ferry control [2]. We will focus on this policy in the sequel and examine its different forms and performances at different levels of control.

3.2 Applying POMDP to Multi-Ferry Control

The use of POMDP is natural for the problem under consideration. Under the DAS architecture, the goal of multi-ferry control is to jointly design global dispatch controller and local search controller such that the overall (discounted) throughput is optimized. The challenge is that both controllers face a dynamic environment that changes over time, e.g., the search controller relies on node distributions, which evolve due to node mobility, and the dispatch controller relies on (predicted) contact processes as well as traffic demands for each domain. Moreover, characteristics of this dynamic environment are only partially observable to the controllers due to limited ferry-node communication range. POMDP is well suited for addressing these challenges using its model of dynamic systems under partial observability.

The local and the global controllers have inherent connections: the global action affects the local control because the number of ferries dispatched to a domain will largely...
determine how long it takes to contact the node; the local performance also affects the global control because the dispatch controller needs to weigh the impact of dispatching more/fewer ferries to each domain to make a balanced decision. The other parameters such as domain sizes, node mobility models, and traffic rates all play a role in the performance. How can we capture all these factors in the POMDP framework? What reward functions should we use for local control as a POMDP and establish lower and upper bounds on the optimal reward, where the lower bound is provided by the myopic policy. Based on the bounds, we discuss conditions under which the myopic policy is close to optimal.

4. LOCAL CONTROL: SEARCH

Consider a domain $d$ to which $k \in \{0, \ldots, K\}$ ferries have been dispatched in the current round; assume $k \leq N$, size of the domain (subscript $d$ is dropped). Given initial node distribution $b_0$ and mobility model $P$, the goal of the local controller $\pi_l$ is to jointly navigate the $k$ ferries to contact the node and deliver traffic. Moreover, due to the discount in throughput calculation, it is intuitively desirable to make the contact as soon as possible. In this section, we cast the problem of local control as a POMDP and establish lower and upper bounds on the optimal reward, where the lower bound is provided by the myopic policy. Based on the bounds, we discuss conditions under which the myopic policy is close to optimal.

4.1 Local Control as a POMDP

In the language of POMDP, the problem can be formulated as follows:

1. **Local state:** $x_t^i = (s_t, \delta_t)$, where $s_t \in S$ is the node’s location at time $t$, and $\delta_t \in \{0, 1\}$ indicates whether the next contact will be the first in the current round ($\delta_t = 1$) or not ($\delta_t = 0$);

2. **Local action:** $a_t \subseteq S$ with $|a_t| = k$ denotes the set of cells for the ferries to cover in slot $t$ (no two ferries cover the same cell);

3. **Local observation:** $d_t = z_t \in \{0, 1\}$ is a joint contact indicator, i.e., $z_t = I_{s_t \in a_t^i}$, where $I$ denotes the indicator function;

4. **Local reward:** the one-time reward $r_t(x^i, a^i, d^i) = \delta z$ gives a unit reward for the first contact, after which $\delta$ transits to 0 and no further reward occurs; the overall reward $R_{\infty}^{\pi_l}$ is defined as in (1) $\pi_l$.

We elaborate on the preceding POMDP formulation: let $\Upsilon^s$ denote the time till (the first) contact (TTC) under policy $\pi_l$. It is easy to see that $R_{\infty}^{\pi_l} = E[\Upsilon^s]$.

4.2 Local Control Policy

The true state $x_t^i$ is not directly known to the controller since node location $s_t$ is unknown. Instead, the controller observes another state $y_t^i$, which consists of the belief $b_t := \Pr(s_t = s|a_t^1, \ldots, a_t^i, \delta_{i-1})$ and the indicator $\delta_t$. At the end of the slot, the new state $y_{t+1}^i = (b_{t+1}, \delta_{t+1})$ transits as

$$
\begin{align*}
\begin{cases}
    b_{t+1} &= P^s(z_t e_a + (1 - z_t) b_{t\setminus a_t}), \\
    \delta_{t+1} &= \delta_t (1 - z_t),
\end{cases}
\end{align*}
$$

where $e_a$ or $b_{t\setminus a_t}$ is the updated belief based on observation $z_t$, and multiplying it by the transition matrix $P^s$ gives the predicted belief for the next slot. The myopic search policy is the one that maximizes the probability of immediate contact:

$$
\pi_{MV}^l(b) = \arg \max_{a^l} \sum_{s \in a^l} b(s),
$$

i.e., the $k$ ferries will search the cells with the top $k$ probabilities in the belief.

4.3 Performance of Local Control

For a given search policy $\pi$ (subscript $l$ dropped for simplicity), let $\Upsilon^s$ denote the TTC under $\pi$ for a given initial belief $b_0$ and $k$ ferries. The distribution of the random variable $\Upsilon^s$ can be characterized as follows. Conditioned on the event that no contact has occurred before $t$ (i.e., the previous $t - 1$ slots are misses), let $a_t^s$ and $\pi_t^s$ denote the action and the belief under policy $\pi$ in slot $t$, and $\rho_t^s$ the conditional probability of contact. By definition, we have $\rho_t^s = \sum_{s \in a_t^s} b_t^s(s)$, and $a_t^s$, $b_t^s$ can be computed from the following updates: starting from $b_0^s = P^s b_0$,

$$
\begin{align*}
a_t^s &= \pi(b_t^s), \quad b_{t+1}^s = P^s b_t^s a_t^s, \quad t = 1, 2, \ldots
\end{align*}
$$

Based on $\rho_t^s$, it is easy to see that the distribution of $\Upsilon^s$ and the corresponding $R_{\infty}^{\pi}$ are given by

$$
\begin{align*}
\Pr(\Upsilon^s = t) &= \rho_t^{s-1} \prod_{j=1}^{t-1} (1 - \rho_j^s), \\
R_{\infty}^{\pi} &= \sum_{t=1}^{\infty} \beta^t \Pr(\Upsilon^s = t) = \sum_{t=1}^{\infty} \beta^t \rho_t^{s-1} \prod_{j=1}^{t-1} (1 - \rho_j^s).
\end{align*}
$$

In particular, plugging in $\rho_t^s = \pi_{MV}^l(b_t^s)$ for $\pi_{MV}^l$ in (5) yields the reward of myopic policy $R_{\infty}^{\pi_{MV}}$.

Although the above method will give the reward for any given policy, it does not indicate how close its performance is to the optimal. Characterization of the optimal reward $R_{\infty}^{\pi_{MV}}$ is an open question, since it is computationally intractable to compute the optimal policy. Hence, we derive the following lower and upper bounds on the optimal reward. The lower bound comes naturally from the myopic policy. For the upper bound, consider the passive belief transitions without any observation: $b^{(l)} := (P^s)^l b_0$. Let $B_{l,k} := \max_{|a|=k} \sum_{s \in a} b^{(l)}(s)$ be the sum of the $k$ largest elements in $b^{(l)}$.

\[\text{Here } P^T \text{ denotes matrix transpose, } e_a \text{ the unit vector with one in the } a \text{th element, and } b_{\alpha} \text{ the posterior belief after a miss given by } b_{\alpha}(s) = 0 \text{ for all } s \in a \text{ and } b_{\alpha}(s) = b(s)/(\sum_{s' \not\in a} b(s')) \text{ otherwise.}\]
Define
\[ R_{\infty} := \beta_0 \left( 1 - \sum_{t=1}^{T_0-1} B_{t,k} \right) + \sum_{t=1}^{T_0-1} \beta_t B_t,k, \]
where \( T_0 := \inf\{ t \geq 1 : 1 - \sum_{j=1}^{t-1} B_{j,k} \leq B_{t,k} \} \). We have the following result.

**Theorem 4.1.** The reward \( R_{\infty}^\pi \) of the optimal search policy satisfies: \( R_{\infty}^\pi \leq R_{\infty} \leq \overline{R}_{\infty} \).

To prove the theorem, we introduce the following lemmas; their proofs can be found in Appendix.

**Lemma 4.2.** For any policy \( \pi \), its reward \( R_{\infty}^\pi \) is monotone increasing with the conditional contact probability \( p_t^\pi \).

**Lemma 4.3.** For any \( t \) and any \( \pi \),
\[ p_t^\pi \leq \frac{B_t,k}{\max(1 - \sum_{j=1}^{t-1} B_{j,k}, B_t,k)} \leq \overline{p}_t. \]

**Proof:** (Theorem 4.1) The lower bound holds trivially. For the upper bound, Lemma 4.2 implies that \( \overline{p}_t \) in Lemma 4.3 into (8) will give an upper bound on the reward:
\[ R_{\infty}^\pi \leq \sum_{t=1}^{T_0} \beta_t \overline{p}_t \prod_{j=1}^{t-1} (1 - \overline{p}_j) \]
\[ = \beta_0 \left( 1 - \sum_{t=1}^{T_0-1} B_{t,k} \right) + \sum_{t=1}^{T_0-1} \beta_t B_t,k = \overline{R}_{\infty}, \]
which implies \( R_{\infty}^\pi \leq \overline{R}_{\infty} \) as the bound holds for any \( \pi \). Note that \( \prod_{j=1}^{t-1} (1 - \overline{p}_j) = 1 - \sum_{j=1}^{t-1} B_{j,k} \).

Closed-form bounds can be obtained by further bounding \( R_{\infty}^{\pi_0} \) and \( \overline{R}_{\infty} \) (see Appendix for proofs).

**Corollary 4.4.** For \( k \) ferries and a domain of size \( N \) (cells), \( R_{\infty}^{\pi_0} \geq \beta k / [N(1 - \beta) + \beta k] \).

In general, there is no non-trivial closed-form upper bound, i.e., \( R_{\infty}^\pi \) can approach \( \beta \) arbitrarily. Under certain conditions, however, we have the following result.

**Corollary 4.5.** If the initial belief is the steady-state distribution \( \mathbf{b}_* \), \( \{ \mathbf{b}_* = \mathbf{P}^T \mathbf{b}_*, \} \), then
\[ \overline{R}_{\infty} = \beta_0 \left[ 1 - B_{*,k}(T_0 - 1) \right] + \frac{B_{*,k}(\beta - \beta_0)}{1 - \beta}, \]
where \( B_{*,k} := \max_{|a|=k} \sum_{s \in a} b_*(s) \) and \( T_0 = \left[ B_{*,k}^{-1} \right] \).

Moreover, if \( \mathbf{P} \) is doubly stochastic, then the above reduces to
\[ \overline{R}_{\infty} \leq \beta k / [(1 - \beta) N]. \]

**Remark:** For a large domain \( N \gg k \), Corollary 4.4 implies that \( R_{\infty}^\pi \geq \beta k / [(1 - \beta) N] \), achievable by the myopic search policy. The worst case is when the node moves equal likely in all directions (i.e., \( \mathbf{P} \) is doubly stochastic) and the initial distribution is uniform, under which Corollary 4.5 says \( \beta k / [(1 - \beta) N] \) is also an upper bound. In this case, the optimal reward \( R_{\infty}^\pi \approx \beta k / [(1 - \beta) N] \). Note that the

5. **GLOBAL CONTROL: DISPATCH**

The global control operates on top of local control at the macro time scale of rounds. Given local search policies, the goal of the global dispatch policy is to allocate the ferries among the domains at the beginning of each round to maximize the total throughput. In this section, we formulate the global control as another POMDP, based on which we develop an approximate myopic dispatch policy with a significantly lower complexity than the original and analyze its performance.

5.1 **Global Control as a POMDP**

The global control problem can be formulated as the following POMDP:

1. **Global state:** \( x_t^r = (s_t, m_t) \), where \( s_t = (s_{t+1})_{t \leq 0} \) still denotes node locations in each domain, and \( m_t = (m_{t+1})_{t \leq 0} \) denotes the buffer state of the BS, where \( m_{t+1}(d) \) is the amount of traffic generated for domain \( d \), all at the beginning of round \( \tau \) (i.e., at time \( (\Delta + 1)(\tau - 1) \));
2. **Global action:** \( a_t^r = (a_t^r)_{t \leq 0} \) represents the distribution of the ferries among the domains in round \( \tau \) (i.e., \( a_t^r(d) \in \mathbb{N} \) and \( \sum_{d=1}^{D} a_t^r(d) = K \));
3. **Global observation:** \( \omega_t^r = (\omega_t, b_t^r) \), where \( \omega_t = (\omega_{t+1})_{t \leq 0} \) still denotes the TTCs for each domain, and \( b_t^r = (b_t^r)_{t \leq 0} \), the updated beliefs of node locations at the end of round \( \tau \) (at time \( (\Delta + 1)\tau \)); if there is no contact with a domain either because no ferry serves the domain \( (a_t^r(d) = 0) \) or because the ferries fail to contact the node within time \( \Delta \), define \( \omega_t \equiv \infty \);
4. **Global reward:** the one-time reward is defined as \( r_\pi(x^r, a^r, \omega^r) := \sum_{d=1}^{D} \beta \Delta \sum_{t=1}^{T_0} t B_{t,k} + \sum_{t=1}^{T_0-1} \beta_t B_t,k \). where the overall reward has a slightly different form from (1):
\[ R_{\infty}^\pi \equiv \mathbb{E} \left[ \sum_{t=1}^{T_0} \beta \Delta \sum_{t=1}^{T_0} t B_{t,k} + \sum_{t=1}^{T_0-1} \beta_t B_t,k \right] \]
We now explain the meaning of this reward function.

**Lemma 5.1.** The global reward \( R_{\infty}^\pi \) is equal to the discounted total throughput \( \lambda_w \) (see Section 2.3) under the dispatch policy \( \pi_w \) and the associated search policies.
The problem is that even the myopic policy can become constraint, whose distribution is given by (7) for initial node $d$. Let $\beta\{\}$ actions. The $|A|$ control due to the large solution space with compute $\mathbb{E}[\beta^v(d)]$ at local level via ferry navigation $\alpha^e$ in each domain (under given $\alpha^e$). Note that the latter is consistent with the local reward $\mathbb{E}[\beta^v(d)]$ in each domain (under given $\alpha^e$). Here the period $\Delta$ is introduced to facilitate synchronization among ferries. Its selection involves a tradeoff: a smaller $\Delta$ means lower probabilities of contact per round but more rounds during a given time, whereas a larger $\Delta$ means higher probabilities of contact per round but fewer rounds. The exact value can be tuned to optimize the overall performance (see Fig. 8 and 11).

5.2 Global Control Policy

Let $y^g$ denote the observed state. Since one part $y^g$ of the state $s^g$ is partially observable and the other part $m^d$, is completely observable, we have a mixed state $y^g := (b^d, m^d)$, where $b^d = (b^d_r)_{r=1}^D$ are the beliefs of node locations $s^g$, as reported by the local controllers. At the end of each round, the new beliefs $b^d_{r+1}$ will go through a sequence of transitions depending on the actions and observations of the local controllers by (4). Instead of repeating those details, we simply consider the output $b^d_r$ as part of the global observation such that $b^d_{r+1} = b^d_r$. The BS buffer state $m^d$ transmits differently for each domain depending on whether the domain receives service or not:

$$m^d_{r+1}(d) = \lambda_d(\Delta + 1) + m^d_r d > \Delta. \quad (14)$$

The global control is much harder to solve than the local control due to the large solution space with $|A| = (K + D - 1)$ actions. The myopic dispatch policy is given by:

$$\pi^{\text{MV}}_g(y^g) := \arg\max_{a^g} \sum_{d=1}^D m(d) \mathbb{E}[\beta^{v(d)}]. \quad (15)$$

Let $Y(d)$ denote the TTC in domain $d$ without deadline constraint, whose distribution is given by (7) for initial node distribution $b_0$ and $\alpha^g(d)$ ferries. By definition, we can compute $\mathbb{E}[\beta^v(d)]$ as a truncated average $\sum_{t=1}^\Delta \beta^t \Pr\{Y(d) = t\}$. The problem is that even the myopic policy can become complex if the action space is large. To address this issue, we look at simpler approximations. First, we note that the global reward is related to the local reward as follows.

**Lemma 5.2.** For each domain with $\alpha^g(d) > 0$ and local control reward $R^d_{\alpha^g}$, $R^d_{\alpha^g} - \beta^\Delta \leq \mathbb{E}[\beta^v(d)] \leq R^d_{\alpha^g}$.

**Proof:** See Appendix. \[ \]

Assuming myopic search policies are used for local control, from the previous analysis (Corollary 4.4), we can replace $R^d_{\alpha^g}$ by its lower bound $R^d_{\alpha^g} \geq \beta^\alpha(d) |1 - \beta^a| N_d + \beta^\alpha(d)$, which gives an approximate myopic dispatch policy:

$$\pi^{\text{MV}}_g(y^g) \approx \arg\max_{a^g} \sum_{d=1}^D m(d) \alpha^{v(d)(1 - \beta^a) N_d + \beta^\alpha(d)}. \quad (16)$$

A nice property of this approximation is that the function $f_d(a) := m(d) a/(1 - \beta^a) N_d + \beta^\alpha(d)$ is increasing and concave in $a$, i.e., extra ferries for a domain only provide diminishing gain compared with existing ferries. This monotonicity means that instead of searching all $(K + D - 1)$ possible values of $\alpha^g$, we can sequentially dispatch one ferry at a time to maximize the reward gain among all the domains. This property implies the following algorithm for implementing (16). Starting from $\alpha^g = 0$, repeat for $k = 1, \ldots, K$:

1. find $a_k := \max_{a^g} \{ m(d) a^g \} = f_d(a^g(d))$.
2. update $a^g(d_k) := a^g(d_k) + 1$.

Compared with the $O(\Delta(D^{K + D - 1}))$ complexity of the myopic dispatch policy, this approximation significantly reduces the complexity to $O(KD)$. In the special case of large domains ($N_d \gg K, \forall d$), we can further simplify (16) into an asymptotically approximate myopic dispatch policy:

$$\pi^{\text{MV}}_g(y^g) \approx \arg\max_{a^g} \sum_{d=1}^D \frac{m(d) a^g}{N_d}, \quad (17)$$

for which the optimal action is simply to dispatch all the ferries to domain $d_k := \arg\max m(d)/N_d$. In contrast, for smaller domains, the ferries may be split among multiple domains in a round. In fact, under (16), they will be dispatched to the same domain if and only if $\exists d$ such that $f_d(K) - f_d(K - 1) \geq f_{d'}(1) - f_{d'}(0)$ for any $d' \neq d$.

5.3 Performance of Global Control

To analyze the performance of global control, it is necessary to characterize the evolution of buffer states $(m^d)_{r=1}^\infty$. For each domain $d$, if $q_r(d)$ denotes the delivery probability in round $r$, then we see from (14) that $(m^d_r)_{r=1}^\infty$ follows a Markovian-stationary process. However, it is not a real Markov chain, as $q_r(d)$ depends on the node beliefs and the policies, making it intractable for analysis. In this section, we derive bounds for the closed-form dispatch policy (17) to obtain insights under the assumption of $N_d \geq K, \forall d$. Under this policy, the delivery probability $q_r(d)$ and the average immediate reward $\mathbb{E}[\beta^{v(d)}]$ can be bounded in closed form as follows.

**Lemma 5.2.** Under myopic search policies and the dispatch policy in (17), let $d_0 := \arg\max m^d(d)/N_d$ denote

$$q_r(d) \leq \frac{1 - q_r(d) - 1 - q_r(d) - \lambda_d}{\lambda_d \Delta} \ldots q_r(d)$$

**Figure 3:** Evolution of buffer state $m^d_r$.
10.5. The domain served in round $\tau$. We have

$$\begin{align*}
q_r(d_r) & \geq 1 - \left(1 - \frac{K}{N_{d_r}}\right)^{\Delta} =: q_r, \\
\mathbb{E}[\beta^{-r}(d_r)] & \geq \frac{\beta K \left(1 - \beta^\Delta \left(1 - \frac{K}{N_{d_r}}\right)^\Delta\right)}{N_{d_r} - \beta (N_{d_r} - K)}.
\end{align*}$$

Proof: See Appendix.

These bounds suggest the following evolution of $m_r$:

$$m_{r+1} = \begin{cases} m_r + \lambda(\Delta + 1) - m_r(d_r) e_{d_r}, & \text{w.p. } q_r, \\ m_r + \lambda(\Delta + 1), & \text{o.w.} \end{cases}$$

(20)

Since $d_r$ is a function of $m_r$, we see that the process $\{m_r\}_{r=1}^\infty$ evolving (20) is Markovian. This chain also determines the reward: since the expected immediate reward is $m_r(d_r) \mathbb{E}[\beta^{-r}(d_r)]$, applying (19) gives a lower bound on the reward

$$r(m_r) := \frac{\beta K m_r(d_r) \left(1 - \beta^\Delta \left(1 - \frac{K}{N_{d_r}}\right)^\Delta\right)}{N_{d_r} - \beta (N_{d_r} - K)},$$

which is only a function of $m_r$. These results lead to the following performance bound.

**Theorem 5.4.** Under myopic search policies and the approximate myopic dispatch policy in (17), the discounted total throughput is lower bounded by

$$R_\infty^{\beta, g} \geq \mathbb{E}\left[\sum_{t=1}^\infty \beta^{(\Delta+1)(r-1)} r(m_r) | m_1 = 0\right] :=: R_\infty^g,$$

(22)

where the expectation is over the Markov chain $\{m_r\}_{r=1}^\infty$ specified in (20).

**Proof:** See Appendix.

Generally, (22) does not have a closed-form solution as it depends on the transient statistics of the Markov chain. Hence, we develop a numerical method to approximate it arbitrarily. Let $M_r := \{(m_r^{(i)}, p_r^{(i)})\}$ denote the set of all possible values of $m_r$ and the probability $p_r$ of reaching it from the initial buffer state $m_1$. Then Algorithm 1 computes a $T$-step approximation of $R_\infty^{\beta, g}$ by computing $M_r$ ($r = 1, \ldots, T + 1$) iteratively. In each iteration (lines 2–8), it enumerates all elements of $M_r$ (line 4), computes the possible next states and their probabilities in $M_{r+1}$ (line 7), and accumulates the corresponding reward (line 8).

**Algorithm 1** Evaluate Global Reward

- **Require:** Domain size $(N_d^*)_{d=1}^\infty$, data rate $\lambda$, number of ferries $K$, round length $\Delta$, discount factor $\beta$, initial BS buffer state $m_1$, and horizon $T$.

- **Ensure:** Return the $T$-step approximated reward lower bound $R_\infty^{\beta, g}$.

1: $M_1 \leftarrow \{(m_1, 1)\}$, $R_\infty^{\beta, g} = 0$

2: for $r = 1$ to $T$

3: $M_{r+1} \leftarrow 0$, $R_\infty^{\beta, g} \leftarrow R_\infty^{\beta, g}$

4: for all $(m_r, p_r) \in M_r$

5: $d_r \leftarrow \arg\max_d \frac{\sum_{i=1}^d \lambda_i}{\sum_{i=1}^d \lambda_i}$

6: $q_r \leftarrow 1 - \left(1 - \frac{K}{N_{d_r}}\right)^\Delta$

7: add $(m_r + \lambda(\Delta + 1) - m_r(d_r) e_{d_r}, p_r q_r)$, $(m_r + \lambda(\Delta + 1), p_r (1 - q_r))$ to $M_{r+1}$

8: $R_\infty^{\beta, g} \leftarrow R_\infty^{\beta, g} + \beta^{(\Delta+1)(r-1)} p_r r(m_r)$

6. SIMULATION AND COMPARISON WITH PREDETERMINED CONTROL

Having developed the DAS policies, we now evaluate their performance against appropriate benchmarks. A benchmark of particular interest is the class of predetermined control policies specified by fixed ferry routes. In the sequel, we will derive the optimal predetermined policies for local and global control and compare them with the proposed policies.

6.1 Local Control

For the predetermined local controller, knowledge of node location is fixed at the steady-state distribution $b_\ast$. Thus, to maximize the chance of contact, the best policy for $k$ ferries is to wait for the node at the $k$ most probable cells $s^{(i)}$ ($i = 1, \ldots, k$) such that $b_\ast(s^{(1)}) \geq b_\ast(s^{(2)}) \geq \ldots$. We will call this the **waiting policy**.

We now compare the waiting policy with the myopic search policy. Suppose the node follows a 2-D random walk on a grid with the level of mobility controlled by an **activeness parameter** $\alpha := \sum_j P(i, j)$ (i.e., probability of moving to a different cell), starting from the steady-state distribution (uniform distribution). As illustrated in Fig. 4–5, the performance of both policies improves (reward increases while TTC decreases) as the number of ferries $k$ increases, and the myopic policy clearly outperforms the waiting policy. The performance gap, however, varies depending on the level of mobility, with a larger gap in low mobility cases. Intuitively, this is because the waiting policy relies on node mobility to create contact opportunities, whereas the myopic policy will actively search for the node and is thus less affected. We also plot the lower and upper bounds on the myopic reward (Corollary 4.4–4.5) and the corresponding bounds on its mean TTC; we note that both bounds track the actual performance consistently and that the bounds are fairly tight.

We have also conducted similar simulations under biased mobility (Fig. 6–7), modeled by a **localized random walk with tightness parameter** $\eta$ [10]. This model allows non-uniform transition probabilities $P(i, j) \propto e^{-\eta \left|j - b\right|}$ biased toward a home cell $h$ (if $\eta \geq 0$), where $\left|j - b\right|$ denotes the taxicab distance between cells $j$ and $h$, and $\eta$ controls the level of bias ($\eta = 0$ for standard random walk). In the simulations, the home cell was at the center of the grid. The results indicate that performance trends and comparisons are similar.
the beliefs of node locations are fixed at their steady states, to that of dynamic global control in Section 5.1 except that

6.2 Global Control

Figure 6: Myopic search policy vs. waiting policy: low-mobility, localized random walk ($\alpha = 0.1$, $\eta = 0$, $N = 25$, $\beta = 0.9$, 5000 Monte Carlo runs).

(a) Local reward. (b) Mean TTC.

Figure 5: Myopic search policy vs. waiting policy: high-mobility, localized random walk ($\alpha = 0.9$, $\eta = 0$, rest as in Fig. 4).

(a) Local reward. (b) Mean TTC.

Figure 7: Myopic search policy vs. waiting policy: high-mobility, localized random walk ($\alpha = 0.9$, $\eta = 0.5$, rest as in Fig. 4).

(a) Local reward. (b) Mean TTC.

where $v^\theta(d)$ is the TTC in domain $d$ under the waiting policy, starting from the steady-state distribution. After one round, the expected buffer state will evolve as $\overline{m}_{\tau+1}(d) = \lambda_d(\Delta + 1) + m_\tau(d)Pr\{v^\theta(d) > \Delta\}$. A crucial difference between (24) and (15) is that $E[\beta^\theta(d)]$ and $Pr\{v^\theta(d) > \Delta\}$ are fixed for any given $\alpha^\theta(d)$. Therefore, given the initial buffer state $\overline{m}_1 = m_1$, the entire sequences of $(\overline{m}_r)_{r=1}^\infty$, and $(\alpha_r^\theta)_{r=1}^\infty$ are determined.

To complete the policy specification, we need to evaluate $E[\beta^\theta(d)]$ and $Pr\{v^\theta(d) > \Delta\}$ for a given $\alpha^\theta(d) = k \in \{1, \ldots, K\}$. Let $H_k$ (index $d$ is dropped for simplicity) denote the hitting time for the $k$ most probable cells $\{s^{(i)}\}_{i=1}^k$ in the steady state $b_*$. starting from $b_*$. Then $E[\beta^\theta(d)] = \sum_{t=1}^{\Delta} \beta(t) Pr\{H_k = t\}$ and $Pr\{v^\theta(d) > \Delta\} = Pr\{H_k > \Delta\}$. By analysis similar to that in Section 4.3, we have $Pr\{H_k = t\} = p_0^\theta \prod_{i=1}^{t-1} (1 - p_0^\theta)$, where $p_0^\theta$ is the conditional contact probability given by $p_0^\theta = \sum_{i=1}^{k} b_0^\theta(s^{(i)})$, for $b_0^\theta = b_*$, and $b_{t+1}^\theta = \mathbf{P}^\theta b_t^\theta |(s^{(i)})_{t=1}^k$.

We now compare the performance of the above predetermined policy with that of the myopic dispatch policy (15) and its approximations (16–17). We assume 2-D random walk for each domain with identical size, activeness, and traffic rate. We first evaluate the performance of these policies for different values of the round length $\Delta$; see Fig. 8 (“approx myopic 1°” for (16) and “approx myopic 2°” for (17)), where for each $\Delta$, we simulate the policies for enough time ($\geq 100$ slots) to approximate the total reward $R_\infty^\alpha$. Interestingly, although it takes much longer to guarantee a contact (at least 5 slots even if nodes do not move), all the dispatch policies achieve their best performance at a smaller $\Delta$: $\Delta = 3$ for the myopic policy (15) and its approximation (16), $\Delta = 2$ for its other approximation (17), and $\Delta = 1$ for the predetermined policy (24).

Based on the above results, we compare the policies under their best $\Delta$ with respect to (wrt) the discounted throughput $\lambda_\beta,T := E[\sum_{i=1}^T N(t)\beta(t)]$ (Fig. 9), where $N(t)$ denotes the amount of traffic delivered in slot $t$, and wrt the undiscounted throughput $\lambda_{\beta=1,T}$ (Fig. 10). The myopic policy significantly outperforms the predetermined policy on both metrics (by 125% on $\lambda_\beta,T$ and by 47% on $\lambda_{1,T}$). Moreover, the two approximations provide close-to-myopic performance at much reduced complexities, where (16) even outperforms the myopic policy on $\lambda_\beta,T$. This shows that myopic policy is suboptimal for global control. We also evaluate an $\epsilon$-approximation (with $\epsilon = 0.25$) of the analytical bound in (22) by Algorithm 1 (“myopic lower bound”), which gives a conservative lower bound for myopic policies.

We repeat the above simulations for a heterogeneous case to those under symmetric mobility, but with a smaller gap between the two policies, as biased mobility provides more information to the waiting policy through its non-uniform steady-state distribution. The analytical bounds are looser in these cases.

In all of the above simulation results, we observe that the reward and the TTC metrics both improve monotonically as the number of ferries increases, but the marginal gain becomes smaller, particularly in the case of TTC; this is in accordance with the theoretical results developed in Section 4.3.

6.2 Global Control

The problem of predetermined global control is analogous to that of dynamic global control in Section 5.1 except that the beliefs of node locations are fixed at their steady states, and the BS buffer state is replaced by its expectation $\overline{m}_r$. For fair comparison with the myopic dispatch policy (15), we consider the following myopic predetermined dispatch policy:

$$\pi^\theta_\alpha(\overline{m}) = \arg \max_{\pi^\alpha} \sum_{d=1}^D \overline{m}_r(d)E[\beta^\theta(d)], \quad (24)$$

where $\pi^\theta_\alpha(\overline{m}) = \arg \max_{\pi^\alpha} \sum_{d=1}^D \overline{m}_r(d)E[\beta^\theta(d)]$. We repeat the above simulations for a heterogeneous case...
of different mobility parameters and traffic rates for each domain under the localized random walk model defined in Section 6.1; see Fig. 11–13 (the optimal \( \Delta \) is 2 for the myopic policies and 1 for the predetermined policy). The results show similar trends, except that the performance gap between the myopic policies and the predetermined policy shrinks (the myopic policy outperforms the predetermined policy by 68% on \( \lambda \), \( T \) and by 26% on \( \lambda, T \)). Intuitively, this is because in heterogeneous cases, prior knowledge such as domain sizes, mobility parameters, and traffic rates already provides good distinction between domains, which helps the predetermined dispatch policy just as biased mobility helps the waiting policy in local control.

7. CONCLUSION

We have considered the control of multiple data ferries in partitioned wireless networks. Compared with the literature, our solution deals with a more challenging case of stochastically moving nodes and limited ferry-node and ferry-ferry communication ranges. Using the approach of stochastic control, we develop a fully dynamic solution via the hierarchical policy of “dispatch” and “search”, which only requires local cooperation between ferries and provides substantial performance improvements over existing predetermined control in high uncertainty scenarios (e.g., uniform node steady-state distributions, identical partitions).

Although we have focused on data dissemination in presenting the detailed solution, our approach is applicable to other communication scenarios as well. Under other communication modes, the local control remains the same whereas the global control needs to be modified. For example, a global POMDP similar to that in Section 5.1, with \( m(d) \) denoting the traffic generated by domain \( d \) and reward \( r_j(x^g, a^g, o^g) := \sum_{d=1}^{D} m(d) \beta^{d+1} I_{\delta(d) \leq \Delta} \), corresponds to data harvesting (with delayed pickup). A more complicated variation can model peer-to-peer communication via the relay of BS by tracking the buffer states of each source domain and the BS with a set of state variables and using the BS buffer state \( m_j \) to compute the reward \( r_j(x^g, a^g, o^g) \) as in Section 5.1. Detailed studies are left to future work.

8. REFERENCES


APPENDIX

Proof of Lemma 4.2

Based on (8), we have

\[
\frac{\partial R^*_{\pi}}{\partial p^*_j} = \beta^j \prod_{j=1}^{t-1} (1 - p^*_j) - \sum_{k=t+1}^{\infty} \beta^k p^*_j \prod_{j=1, j \neq t}^{k-1} (1 - p^*_j) \\
= \left[ \prod_{j=1}^{t-1} (1 - p^*_j) \right] \left( \beta^t - \mathbb{E}[\beta^T \mid T > t] \right) \geq 0.
\]

Therefore, \( R^*_{\pi} \) is monotone increasing with \( p^*_j \).

\[ \square \]
Proection of Lemma 4.3

First, we prove by induction that (recall \( b^{(t)} = (P^T)^t b_0 \))

\[
b^*_t(i) \leq \frac{b^{(t)}(i)}{1 - \sum_{s\in\mathcal{A}_t^j} b^{(j)}(s)}, \quad \forall i \in \mathcal{S}, \ t \geq 1.
\]  

(25)

For \( t = 1 \), \( b^*_1(i) = P^T b_0 = b^{(1)}(i) \). For \( t > 1 \),

\[
b^*_t(i) = \frac{b^{(t)}(i)}{1 - \sum_{s\in\mathcal{A}_t^j} b^{(j)}(s)}, \quad \forall i \in \mathcal{S}, \ t \geq 1.
\]

(25)

obtained by applying (25) for \( b^*_{t-1}(j) \). This proves (25).

Then, we apply the above to \( p_t^* = \sum_{s\in\mathcal{A}^t_i} b^*_t(i) \):

\[
p_t^* \leq \frac{\sum_{s\in\mathcal{A}^t_i} b^{(t)}(s)}{1 - \sum_{j=1}^{t-1} \sum_{s\in\mathcal{A}^{t-1}_j} b^{(j)}(s)} \leq \overline{p}_t
\]

for \( \overline{p}_t \) defined as in (10).

Proof of Corollary 4.4

The proof is based on the fact that \( p_t^{MV} \geq k/N, \ \forall t \). By Lemma 4.2, we will lower bound \( R_{\infty}^{MV} \) by replacing \( p_t^{MV} \) with \( k/N \), which yields

\[
R_{\infty}^{MV} \geq \sum_{t=1}^{\infty} \beta^t \frac{k}{N}(1 - \frac{k}{N})^{t-1} = \frac{\beta k}{N(1 - \beta)} + \frac{\beta k}{N(1 - \beta)^2}.
\]

(26)

Proof of Corollary 4.5

If \( b_0 = b_* \), then \( b^{(t)}(i) = b_0(i), \ \forall t \). Accordingly, \( B_{t,k} \equiv B_{s,k} \), and \( T_0 = [B_{s,k}^{-1}] \). Plugging these into (9) gives (11).

If, in addition, \( P \) is doubly stochastic, then it is known that \( b_* \) is uniform, i.e., \( B_{s,k} = k/N \), applying which to (11) gives

\[
\bar{R}_{\infty} = \beta^{\lfloor N/k \rfloor} \left[ 1 - \frac{k}{N} \left( \frac{N}{k} - 1 \right) \right] + \frac{k(\beta - \beta^{\lfloor N/k \rfloor})}{N(1 - \beta)}.
\]  

(26)

If we maximize the right-hand side of (26) with respect to \( \lfloor N/k \rfloor \), calculation shows that the maximum is achieved at \( \lfloor N/k \rfloor = \frac{N}{\log \beta}, \frac{N}{\log \beta} + 1 \), or \( \frac{N}{\log \beta} - 1 < \frac{1}{\log \beta} + 1 \). At the first two values, the right-hand side is

\[
\frac{k\beta(1 - \beta^{N/k})}{N(1 - \beta)} < \frac{\beta k}{(1 - \beta)N}.
\]

at the third value, it is

\[
\frac{k}{N} \left( \frac{\beta^{N+1} - \beta}{\log \beta} + \frac{\beta}{1 - \beta} \right) < \frac{\beta k}{(1 - \beta)N}.
\]

Combining both gives \( \bar{R}_{\infty} < \beta k/(1 - \beta)N \).

Proof of Lemma 5.2

The upper bound holds because \( R_{\infty}^{d_t} = \mathbb{E}[\beta^{\tau(d_t)}] \) while \( \mathbb{E}[\beta^{\tau(d_t)}] \) is a truncated average. The lower bound is because

\[
R_{\infty}^{d_t} - \mathbb{E}[\beta^{\tau(d_t)}] = \sum_{t=1}^{\Delta} \beta^t \Pr\{\tau(d_t) = t\}
\]

\[
\leq \beta^\Delta \Pr\{\tau(d_t) > \Delta\} \leq \beta^\Delta.
\]

(27)

Proof of Theorem 5.4

Due to discount, the total throughput \( R_{\infty}^{d_t} \) is an increasing function of the probability of delivery \( q_t(d_t) \) and is thus lower bounded if \( q_t(d_t) \) is replaced by its lower bound in (18), making \( m_t \) evolve according to (20).

Moreover, for given \( (b_*, m_*) \), the expected immediate reward under the dispatch policy (17) is \( m_t(d_t) \mathbb{E}[\beta^{\tau(d_t)}] \), which is lower bounded by \( r(m_*) \) defined in (21) for any \( b_* \) due to (19). Combining these two facts proves the result.