Hierarchical Mobility Control for Data Ferries under Constrained Message Delays

Liang Ma[†], Ting He[‡], Ananthram Swami[§], Kang-won Lee[‡] and Kin K. Leung[†]
[†]Imperial College, London, UK
[‡]IBM T. J. Watson Research Center, Hawthorne, NY, USA
[§]Army Research Laboratory, Adelphi, MD, USA
[†]{l.ma10, kin.leung}@imperial.ac.uk, [‡]{the, kangwon}@us.ibm.com, [§]ananthram.swami@us.army.mil

***Abstract*—We consider the problem of controlling mobile data ferries for message delivery among disconnected, scattered domains in a highly partitioned network. Existing work on data ferry control mostly focuses on predetermined ferry routes, assuming full observations at the ferry and no explicit Quality of Service (QoS) constraints on the resulting communications. In this paper, we aim at designing a QoS-enabled ferry control solution, which handles both partial observations and bounded message delays. To this end, we extend our previous work on data ferry control with partial observations into a comprehensive hierarchical framework called Switch-and-Navigate (SAN), which consists of a global switch policy for determining the best domain to visit and a local navigation policy per domain for searching for nodes within individual domains. Under the assumption of Markovian node mobility, both the global and the local control problems are formulated as Partially Observable Markov Decision Processes (POMDPs) to maximize the discounted effective throughput over all domains. Due to the fact that the optimal solution to POMDP is PSPACE-hard, we develop heuristic policies and further approximations for efficient computation. Simulation results show that the proposed policies can significantly improve the performance over predetermined alternatives.**

I. Introduction

The demanding requirements of mobile ubiquitous communications have promoted the development of Highly Partitioned Mobile Ad Hoc Networks (HP-MANET) in which the network, self-organized without the aid of any established infrastructures, is partitioned into several permanently disconnected autonomous domains due to physical obstacles, limited radio transmission range, severe environmental conditions, or simply security reasons. Applications of such networks can be found in many challenged environments, such as battlefield operations and disaster relief in large areas. Existing research on Delay/Disruption-Tolerant Networking (DTN) (e.g., [1]) has focused on *intermittently partitioned* networks, assuming the disconnected links will be reconnected or new routes can be discovered, which makes the solutions inapplicable to *permanently partitioned* networks. In these networks, to bridge communications between disconnected domains, designated communication nodes called *data ferries* have been proposed to serve as a carrier to deliver messages from one domain to another. Programmed to move in a predetermined or dynamic pattern, data ferries are capable of self-navigating within and between domains to collect and deliver messages upon contacting regular nodes. Therefore, proper mobility control is needed for data ferries to operate efficiently. A major challenge in applying data ferries to military networks is that military units exhibit high mobility, making it difficult to maintain accurate node information at the data ferry. Moreover, unlike traditional DTN, messages in military operations usually have a finite lifetime, which must be considered in data ferry control. In this paper, we aim to address these challenges by designing control policies under tolerable delay requirements, with the goal of optimizing the throughput performance in the network.

Research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

A. Related Work

Most existing work on data ferry control have concentrated on the stationary or fully observable scenarios, where non-ferry nodes are either stationary or always report their positions when moving. Focusing on the case of intra-domain ferry control, algorithms and prototypes for mobile elements scheduling were proposed in [2, 3]. Solutions in [4, 5] utilized a mobile node to move randomly within a stationary sparse network to deliver data opportunistically. Without requiring global knowledge, authors of [6, 7] took advantage of non-randomness of node movements to design proactive control policies. Their solutions, however, require long-range radio communications and the ability to change (non-ferry) node trajectories, which are not feasible in many applications. The authors of [8, 9] extended the Traveling Salesman Problem (TSP) to reduce the delivery delay, but ignored the fact that buffers in mobile nodes have a diverse collection of messages with non-identical residual lifetimes. A comprehensive framework for dynamic ferry control was introduced in [10]; however, it assumed unit traveling time between any two domains and unbounded message delays.

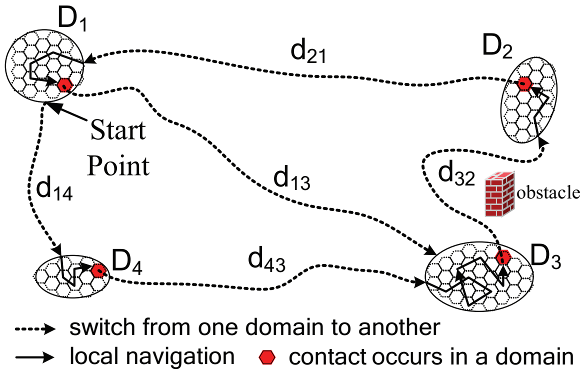


Fig. 1. Example of Switch-and-Navigate: It follows a Switch-Search-Load-Carry-and-Switch cycle

B. Summary of Contributions

In this paper, we address the problem of ferry mobility control under partial observations[1] and finite delay constraints. Assuming Markovian node mobility in each domain, we borrow the control framework of Partially Observable Markov Decision Process (POMDP) [11] used in [10] and extend it to a 2-tier hierarchical policy, called Switch-and-Navigate (SAN), with each layer modeled as a POMDP problem. As illustrated in Fig. 1, a ferry controlled by SAN will search for the gateway (see §II-A) in the current domain according to the "Navigate" policy until contact (the ferry and the gateway are in the same cell), after which the ferry will exchange messages with the gateway and then switch to another domain following the "Switch" policy. In summary, our contributions are:

1) Extended control framework: Compared with [10], our control framework is enriched with two extensions that substantially improve its practical value: finite message lifetime and general inter-domain traveling delays. The first induces a finite life cycle per message, decomposed into waiting time (at source domain) and carrying time (at data ferry). Accordingly, the controller state space is extended to include a (source-domain) gateway buffer state per domain and a ferry buffer state to keep track of the residual lifetimes of pending messages. The second relaxes the unit traveling delay assumption in [10] to a general inter-domain distance (measured by traveling time) matrix, thereby better modeling ferry operations in the scenario of a large field and scattered domains.

2) Hierarchical control policies: Under the framework of SAN, the solution in [10] is borrowed to provide efficient local control (navigate) with only partial observation of the domain. For global control (switch), we develop myopic and two-step policies based on ferry and gateway buffer states, inter-domain distances, and local control policies. Due to the extended controller space, even the two-step policy can be too complicated to compute, and thus we also propose a set of approximations with various performance-complexity tradeoffs.

[1]We will focus on the case of incomplete observation, although our framework also applies to inaccurate observation if the inaccuracy can be modeled probabilistically.

3) Performance evaluation: Our solution is compared with the state-of-art solution of optimized predetermined control. Through extensive simulations, we show that the proposed two-step policy and its approximations outperform the optimized predetermined policy in terms of discounted effective throughput and message loss ratio.

The rest of the paper is organized as follows: Section II formulates the problem. Section III reviews the basis of POMDP, followed by the local and the global control policies in Section IV and V. The optimized predetermined control policy is given in Section VI, after which the simulation results are presented in Section VII. Section VIII concludes the paper.

II. Problem Formulation

A. Network Model

Suppose there are $\widehat{d}$ disjoint domains and only *one* ferry in the network. Let $\mathfrak{D}$ be the set of domain IDs, i.e., $\mathfrak{D} = \{1, ..., \widehat{d}\}$ with Domain i denoted by D_i.

Model of a Single Domain: In each domain, based on the task requirements, the node movement is assumed to follow a Markov chain with known transition matrix $\mathbf{P}_q (q \in \mathfrak{D})$, which will not be affected by the data ferry. To simplify the problem, suppose each domain contains one *gateway* node which serves as the source/destination of all inter-domain messages for this domain. The task of the data ferry is to search for the gateway in each targeted domain and exchange messages. To facilitate control, each domain is divided into $N_q (q \in \mathfrak{D})$ non-overlapping *cell*s which constitute the state space S_q of the Markov chain. The ferry and the gateway can communicate with each other only within the same cell. In this paper, the basic time unit is a *slot*, within which the gateway and the ferry will move among the cells according to the gateway mobility model and the ferry control policy. In the sequel, "at t" is short for "at *slot* t", and t is the global absolute time.

Model of Inter-domain Relationships: Suppose a constant number of $\overline{\lambda}_{i,q}$ $(q \neq i)$ fixed-size messages with lifetime l_{max} are generated for D_q in D_i per slot[2]. We assume that each domain diameter is much smaller than the distance between any two domains, thus the domain layout can be modeled by a distance matrix $\mathbf{d} = (d_{ij})_{i,j \in \mathfrak{D}}$ measured by the number of slots the ferry takes to move from D_i to D_j, whereas each intra-domain movement is assumed to take a single slot.

B. Control Objective

The goal of the data ferry is to determine the best trajectory for delivering as many messages as possible before they expire, jointly considering run-time observations, inter-domain distances, and the number of messages and their residual lifetimes in both the ferry and the gateway buffers. For this purpose, let $\lambda^{\pi}_{\text{total}}$ be the *discounted effective throughput* under policy π:

$$\lambda^{\pi}_{\text{total}} \triangleq \mathbb{E}\big[\sum_{t=1}^{\infty} N^{\pi}(t)\beta^t\big],\ 0 < \beta < 1, \tag{1}$$

[2]This work can be extended to the case of random $\overline{\lambda}_{i,q}$ (only adding complexity to computing (1)).

where $\mathbb{E}[\cdot]$ is the mathematical expectation, β is a given *discount factor* and $N^{\pi}(t)$ is the number of messages delivered within their lifetimes at t. The goal of this paper is to design a policy π to maximize $\lambda^{\pi}_{\text{total}}$.

III. Theoretical Basis: POMDP

The POMDP [11] is a general framework to model the decision process in which a controller dynamically selects actions from a set $\mathcal{A}$ to play on a system with a hidden state drawn from $\mathcal{S}$, which evolves stochastically according to transition probabilities $\mathcal{T}$, to earn reward r (a function of the state and the action). Although it cannot observe the underlying states directly, the controller gets hints on the states from run-time observations in $\mathcal{O}$, related to the states through conditional observation probabilities Ω. The observations allow the controller to maintain a probability distribution of the current state (called belief $\mathbf{b}$) over states $\mathcal{S}$. A POMDP is thus represented as a tuple $(\mathcal{S}, \mathcal{O}, \mathcal{A}, \Omega, \mathcal{T}, r)$, and the goal is to compute a policy π of selecting actions based on belief, i.e., $a = \pi(\mathbf{b}) \in \mathcal{A}$, such that the total discounted reward $R^{\pi} = \mathbb{E}[\sum_{t=1}^{\infty} \beta^t r(\mathbf{b}_t, a_t, o_t)]$ is maximized, where $\beta \in (0,1)$ is a *discount factor* to ensure convergence. It is known from [11] that the optimal policy π^* is the solution to the *Bellman equation* over the horizon T (duration of the control):

$$V_T(\mathbf{b}_t) = \beta \max_{a \in \mathcal{A}} \mathbb{E}\Big[r(\mathbf{b}_t, a, o_t) + V_{T-1}(\mathbf{b}_{t+1})\Big]. \qquad (2)$$

The complexity is, however, PSPACE-hard even for finite T.

The POMDP naturally suits the problem of data ferry control in that it enables us to continuously control the movements of the ferry in a changing environment (mobile gateways, message generation and expiration) without requiring full observation of the environment. In the sequel, we will give detailed POMDP formulations for both the local and the global control and develop efficient policies based on simplifications of (2).

IV. Local Control: Navigate

In this section, we focus on the local control problem: controlling the ferry to find the gateway in a given domain. We only consider D_q ($q \in \mathfrak{D}$), assuming D_q has been selected by the global control policy. In the language of POMDP, the local control can be formulated as:

1) *Local state space*: S_q (cells in D_q). The state $s_t^{(q)}$ ($s_t^{(q)} \in S_q$) means that the gateway of D_q is in cell $s_t^{(q)}$ at t.

2) *Local observation space*: $O_q^l = \{0, 1\}$, where each observation $o_t^{(q)} \in O_q^l$ is an indicator of ferry-gateway contact at t.

3) *Local action space*: Action $a_t^{(q)}$ specifies the destination cell of the ferry at t. Since ferry navigation is over the entire domain, $a_t^{(q)} \in \mathcal{A}_q^l = S_q$.

4) *Observation probability*: Since a contact occurs if and only if the ferry and the gateway are in the same cell, $\Omega_q^l(o_t^{(q)} | s_t^{(q)}, a_t^{(q)}) = 1$ if and only if $o_t^{(q)} = I_{s_t^{(q)} = a_t^{(q)}}$ ($I_{\cdot}$ is the indicator function).

5) *State transition probability*: Under the Markovian mobility with transition matrix[3] $\mathbf{P}_q$, $\mathcal{T}(s_{t+1}^{(q)} | s_t^{(q)}, a_t^{(q)}) = (P_q)_{ij}$ for $s_t^{(q)} = i$ and $s_{t+1}^{(q)} = j$, regardless of ferry action $a_t^{(q)}$ (the motion of the gateway is independent of the ferry control).

6) *Local reward function*: The one-time reward at t is $r_{q,t}^l = 1$ if a contact occurs (we assume that the relevant payload can be exchanged in a single contact) and $r_{q,t}^l = 0$ otherwise.

As it is PSPACE-hard to solve (2), we resort to the *myopic policy* which only considers the one-time reward in the current slot; nevertheless, it shows near-optimal performance in [10]. Let $\mathbf{b}_t^{(q)}$ denote the posterior distribution of the gateway at t just before an observation is taken (i.e., $(b_t^{(q)})_s \triangleq \Pr\{s_t^{(q)} = s | o_1^{(q)}, \ldots, o_{t-1}^{(q)}\}$). Then $\mathbf{b}_t^{(q)}$ is updated as[4]:

$$\mathbf{b}_{t+1}^{(q)} = o_t^{(q)} \mathbf{P}_q^T \mathbf{e}_{a_t^{(q)}} + (1 - o_t^{(q)}) \mathbf{P}_q^T (\mathbf{b}_t^{(q)})_{\setminus a_t^{(q)}}. \qquad (3)$$

Then the *local myopic policy* is shown in (4).

$$\pi_l^{MY}(\mathbf{b}_t^{(q)}) = \arg\max_{a_t^{(q)} \in \mathcal{A}_q^l} \mathbb{E}[r^l(\mathbf{b}_t^{(q)}, a_t^{(q)}, o_t^{(q)})] = \arg\max_{u \in S_q} (b_t^{(q)})_u \qquad (4)$$

V. Global Control: Switch

In this section, we consider the global control problem: selecting the next domain to serve, considering inter-domain distances and traffic demands. We first introduce some notations to describe the extended controller space, and then investigate some heuristic policies and approximations based on (2).

A. Definitions

1) Distance Matrix: $\mathbf{d}$, where $d_{ij} = d_{ji}$, $d_{ii} = 0$ ($i, j \in \mathfrak{D}$). In $\mathbf{d}$, d_{ij} is the traveling distance from D_i to D_j, normalized by the ferry speed to the number of slots (might not meet the *triangle inequality* due to obstacles).

2) Traffic Generation Matrix: Let $\boldsymbol{\lambda}_j$ ($j \in \mathfrak{D}$) be the $\widehat{d} \times l_{max}$ traffic generation matrix of the gateway per slot in D_j (as shown in (5)). In $\boldsymbol{\lambda}_j$, $(\lambda_j)_{i,t}$ is the number of messages destined to D_i from D_j with residual lifetime t, where $(\lambda_j)_{i,l_{max}} = \overline{\lambda}_{j,i}$ (described in § II-A) with $(\lambda_j)_{j,l_{max}} = 0$ and $(\lambda_j)_{i,t} = 0$ for $1 \leq t < l_{max}$ since all new messages have the same lifetime. To ensure feasibility, l_{max} is subject to $l_{max} \geq 1 + \max_{i,j,i \neq j}(d_{ij})$.

$$\boldsymbol{\lambda}_j \triangleq \begin{pmatrix} 0 & 0 & 0 & 0 & \cdots & 0 & (\lambda_j)_{1,l_{max}} \\ 0 & 0 & 0 & 0 & \cdots & 0 & (\lambda_j)_{2,l_{max}} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & (\lambda_j)_{\widehat{d},l_{max}} \end{pmatrix} \qquad (5)$$

3) Gateway and Ferry Buffer State: Let $\mathbf{G}(t) \triangleq \big(\mathbf{G}_1(t), \mathbf{G}_2(t), ..., \mathbf{G}_{\widehat{d}}(t)\big)$ be the vector of gateway buffer states, in which $\mathbf{G}_j(t)$ is the gateway buffer state in D_j at t and $\mathbf{F}(t)$ the ferry buffer state at t, where $\big(G_j(t)\big)_{i,k}$ and $\big(F(t)\big)_{i,k}$ denote the number of messages destined for D_i with residual lifetime k in buffer $\mathbf{G}_j$ and the ferry at t, respectively.

4) First Contact Time (FCT): FCT, represented as $\boldsymbol{\gamma} \triangleq (\gamma_i)_{i=1}^{\widehat{d}}$, is the relative time which starts from the ferry entering into a domain until the first contact with the gateway.

[3]For a vector (matrix) $\mathbf{X}$, X_i (X_{ij}) denotes its ith ((i,j)th) element.

[4]For a vector $\mathbf{x}$, $\mathbf{x}' = \mathbf{x}_{\setminus y}$ means $x'_z = x_z/(1 - x_y)$ for $z \neq y$ and $x'_y = 0$. Let $\mathbf{e}_u$ denote the unit vector with $e_u = 1$.

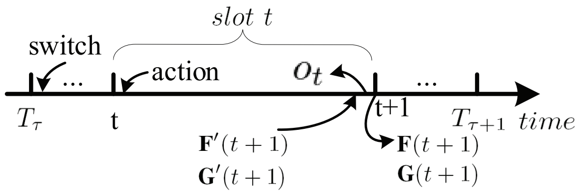


Fig. 2. Update Procedure of Buffer States

5) Inter-contact Time: Represented as ϕ, it is the time between two consecutive contacts among all domains.

6) Round: The duration between two consecutive contacts is called a *round*. Let T_τ denote the start of round τ. According to the definition of the round, T_τ is also the end of round $\tau - 1$. Let ϕ_τ be the inter-contact time of round τ, then $T_\tau = \sum_{i=1}^{\tau-1} \phi_i$.

Based on above notations, the tuple of the global controller state is represented as: $(\mathbf{F}, \mathbf{G}, \mathbf{b}, D_{\mathfrak{L}})$, where $D_{\mathfrak{L}}$ is the domain containing the data ferry at T_τ and $\mathbf{b}$ is the global belief vector with $\mathbf{b} = (\mathbf{b}^{(q)})_{q=1}^{\widehat{d}}$.

B. Bellman Equation for Global Control

The expression for discounted effective throughput, given in (2), can be rewritten as (6), where $\widetilde{T}$ is the total number of rounds in the global control, $\widetilde{V}_{\widetilde{T},i} \triangleq V_{\widetilde{T}}(\mathbf{F}(T_\tau), \mathbf{G}(T_\tau), \mathbf{b}_{T_\tau}, i)$, $\widetilde{V}_{\widetilde{T}-1,j} \triangleq V_{\widetilde{T}-1}(\mathbf{F}(T_{\tau+1}), \mathbf{G}(T_{\tau+1}), \mathbf{b}_{T_{\tau+1}}, j)$ and ζ_{γ_j} denotes the number of delivered messages when a contact occurs at $T_\tau + \gamma_j + d_{ij}$ in D_j.

$$\widetilde{V}_{\widetilde{T},i} = \max_{j, j \neq i} \mathbb{E}\left[\beta^{\gamma_j + d_{ij}}\left(\zeta_{\gamma_j} + \widetilde{V}_{\widetilde{T}-1,j}\right)\right] \tag{6}$$

Since the solution to (6) is PSPACE-hard, *heuristic global policies* are explored in this paper.

C. Update of the global belief vector

Let $\mathbf{b}_0^{(q)}$ be the steady-state distribution of $\mathbf{P}_q$ (the transition matrix under the Markov model). The update of $\mathbf{b}$ starts with $\mathbf{b}_1^{(q)} = \mathbf{b}_0^{(q)}$ ($q \in \mathfrak{D}$). When $t > 1$, (i) If the ferry is in D_q ($q \in \mathfrak{D}$), $\mathbf{b}_t^{(q)}$ is updated as (3); (ii) otherwise, $\mathbf{b}_{t+1}^{(q)} = \mathbf{P}_q^T \mathbf{b}_t^{(q)}$.

D. Update of the Gateway and the Ferry Buffer State

Before discussing the update of all buffer states, we introduce a $l_{max} \times l_{max}$ left shift matrix $\mathbf{H}$ with ones only on the subdiagonal, and zeroes elsewhere. The message residual lifetime in the buffer will subtract 1 per slot, which is equivalent to left shifting of the buffer state matrix.

The gateway and the ferry buffer state experience a two-phase update, i.e., before and after the observation is taken, but both at the end of the slot (illustrated in Fig. 2).

Let $\mathbf{G}^{'}(t+1)$ and $\mathbf{F}^{'}(t+1)$ denote the gateway and the ferry buffer states at $t+1$ before the observation is taken. At the gateway, $\mathbf{G}_j(t)$ ($j \in \mathfrak{D}$) is first left shifted, after which the corresponding traffic generation matrix is added (as shown in (7)). If a contact is observed by the end of slot $t+1$, messages are collected and $\mathbf{G}_j(t+1)$ is reset to $\mathbf{0}$; otherwise, $\mathbf{G}_j(t+1)$ is the same as $\mathbf{G}_j^{'}(t+1)$ (as shown in (8)).

$$\mathbf{G}_j^{'}(t+1) = \mathbf{G}_j(t)\mathbf{H} + \boldsymbol{\lambda}_j, \tag{7}$$

$$\mathbf{G}_j(t+1) = \begin{cases} \mathbf{G}_j^{'}(t+1), & o_t^{(j)} = 0, \\ \mathbf{0}, & o_t^{(j)} = 1. \end{cases} \tag{8}$$

Similarly, $\mathbf{F}(t)$ is first left shifted before the observation is taken at $t+1$ (as shown in (9)). After the observation, if a contact occurs in D_j ($j \in \mathfrak{D}$), the jth row in the ferry buffer state is reset to 0 and new messages are collected from D_j (as shown in (10), where $\mathbf{R}_j$ is the identity matrix except that row j is $\mathbf{0}$).

$$\mathbf{F}^{'}(t+1) = \mathbf{F}(t)\mathbf{H}, \tag{9}$$

$$\mathbf{F}(t+1) = \begin{cases} \mathbf{F}^{'}(t+1), & o_t^{(j)} = 0, \\ \mathbf{R}_j\mathbf{F}^{'}(t+1) + \mathbf{G}_j^{'}(t+1), & o_t^{(j)} = 1. \end{cases} \tag{10}$$

E. Heuristic Global Policies

We now elaborate on the Myopic (MY) and Two-step (TS) Global Policies. Let $p_{i,t}^{\pi_l} \triangleq \max_u (b_t^{(i)})_u$ be the contact probability in D_i at t under the local myopic policy. Then the distribution of γ_i (FCT in D_i) is obtained by $\Pr(\gamma_i = t) = p_{i,t}^{\pi_l} \prod_{m=1}^{t-1}\left(1 - p_{i,m}^{\pi_l}\right)$.

1) Myopic Policy (MY): In the myopic policy, the decision is made to only optimize the discounted amount of messages delivered in the current round. Suppose at T_τ, the ferry is in D_i. Then the ferry calculates how many messages can be delivered to any other domain D_j ($j \in \mathfrak{D} \bigcap j \neq i$):

$$Y_j^{MY} = \mathbb{E}[\beta^{\gamma_j + d_{ij}}\zeta_{\gamma_j}] = \sum_{\gamma_j=1}^{\infty}\left(\Pr(\gamma_j)\beta^{\gamma_j + d_{ij}}\zeta_{\gamma_j}\right), \tag{11}$$

where $\zeta_{\gamma_j} = \sum_{m=1}^{l_{max}}\left(F^{'}(T_\tau + d_{ij} + \gamma_j)\right)_{j,m}$. Therefore, the *global myopic policy* for switching from D_i is:

$$\pi_g^{MY}(D_i) = \arg\max_{j, j \neq i} Y_j^{MY}. \tag{12}$$

2) Two-Step Policy (TS): For the two-step policy, not only messages delivered in the current round, but also those delivered in the next round should be considered. Instead of considering multiple rounds in (6), only two rounds are considered in TS. Suppose the ferry is in D_i at T_τ and in D_j at $T_{\tau+1}$, where $T_{\tau+1} = T_\tau + d_{ij} + \gamma_j$, then

$$\widetilde{V}_{1,j} = \max_{k, k \neq j} \mathbb{E}\left[\beta^{\gamma_k + d_{jk}} \sum_{m=1}^{l_{max}}\left(F^{'}(T_{\tau+1} + d_{jk} + \gamma_k)\right)_{k,m}\right], \tag{13}$$

$$Y_j^{TS} = \mathbb{E}\left[\beta^{\gamma_j + d_{ij}}\left(\zeta_{\gamma_j} + \widetilde{V}_{1,j}\right)\right], \tag{14}$$

where $\widetilde{V}_{1,j}$ denotes $V_1(\mathbf{F}(T_{\tau+1}), \mathbf{G}(T_{\tau+1}), \mathbf{b}_{T_{\tau+1}}, j)$. The *global two-step policy* for switching from D_i is:

$$\pi_g^{TS}(D_i) = \arg\max_{j, j \neq i} Y_j^{TS}. \tag{15}$$

After an actual contact in D_j (selected by (15)). The ferry will follow the above two-step policy to determine the next domain based on the newly updated buffer states and beliefs.

F. Steady-state-based Approximations of Global Policies

In a vast open area, since $\widehat{d}$ and N_q ($q \in \mathfrak{D}$) are relatively large and the inter-domain distance is greater than 1, $\mathbf{b}_t^{(q)}$ will approach $\mathbf{b}_0^{(q)}$ each time the ferry returns to D_q. Therefore, to reduce the complexity of control, we reset the belief to the steady state distribution at the beginning of each round, i.e., $\mathbf{b}_{T_\tau}^{(q)} \equiv \mathbf{b}_0^{(q)}$ ($q \in \mathfrak{D}$), although local belief updates within a round still follows (3). The corresponding steady-state-based myopic and two-step policies are denoted by S-MY and S-TS, respectively.

G. Further Approximations of S-TS

Substantial computation is still required for S-TS. Therefore, several further approximations are investigated: (i) the FCT in D_q is approximated by its average $\overline{\gamma}_q = \sum_{\gamma_q=1}^{\infty} \gamma_q \Pr(\gamma_q)$ under π_l^{MY} (starting from the steady state); (ii) discount β is ignored.

1) S-TS approximation with $\overline{\gamma}$ in the 2nd step (S-TSA^2): For the second step in S-TSA^2, we assume the average FCT $\overline{\gamma}_k$ in D_k is used to calculate V_1:

$$\widetilde{V}_{1,j}^{S-TSA^2} = \max_{k,k\neq j} \sum_{m=1}^{l_{max}} \left(F^{'}(T_{\tau+1} + d_{jk} + \overline{\gamma}_k)\right)_{k,m}. \quad (16)$$

Therefor, the S-TSA^2 policy is

$$\pi_g^{S-TSA^2}(D_i) = \arg\max_{j,j\neq i} \mathbb{E}\left[\left(\zeta_{\gamma_j} + \widetilde{V}_{1,j}^{S-TSA^2}\right)\right]. \quad (17)$$

2) S-TSA^2 with 2nd round to the nearest domain (S-TSA^2-N): If we assume the ferry switches to the nearest domain from D_j in the 2nd round, Eq. 16 is then modified as

$$\widetilde{V}_{1,j}^{S-TSA^2-N} = \sum_{m=1}^{l_{max}} \left(F^{'}(T_{\tau+1} + d_{j\underline{k}} + \overline{\gamma}_k)\right)_{\underline{k},m}, \quad (18)$$

where $\underline{k} = \arg\min_{s,s\neq j} d_{js}$. Then the S-TSA^2-N policy is

$$\pi_g^{S-TSA^2-N}(D_i) = \arg\max_{j,j\neq i} \mathbb{E}\left[\left(\zeta_{\gamma_j} + \widetilde{V}_{1,j}^{S-TSA^2-N}\right)\right]. \quad (19)$$

3) S-TSA^2 with $\overline{\gamma}$ in the 1st step (S-TSA^{1,2}): Like S-TSA^2, S-TSA^{1,2} assumes the FCT in the 1st step is also the average FCT $\overline{\gamma}_j$ in D_j. Then the S-TSA^{1,2} policy is

$$\pi_g^{S-TSA^{1,2}}(D_i) = \arg\max_{j,j\neq i} Y_j^{S-TSA^{1,2}}, \quad (20)$$

where $Y_j^{S-TSA^{1,2}} = \sum_{m=1}^{l_{max}} \left(F^{'}(T_\tau + d_{ij} + \overline{\gamma}_j)\right)_{j,m} + \widetilde{V}_{1,j}^{S-TSA^2}$.

VI. Benchmark: Predetermined Control

For comparison, we examine the alternative approach of predetermined control, where the state-of-the-art solution is Optimized Way-Points (OPWP) [8]. In OPWP, the ferry carefully chooses some way-points and waits at each of these way-points for a fixed number of slots to achieve a specified minimum cumulative contact probability p_i. The trajectory of the ferry is formed by connecting these way-points by a shortest path, given by classical TSP algorithms. In this paper, each domain only contains one gateway and different domains have no overlap. Therefore, OPWP will be revised for this scenario.

Algorithm 1: Way-Points selection in D_i
Initialization: For each cell s in D_i, set $w_s^{(i)} = 0$.
While $\sum_{s\in S_i} \sum_{t=1}^{w_s^{(i)}} \Pr_i(s,t) < p_i$
$s^* = \arg\max_s \Pr_i(s, w_s^{(i)} + 1)$
$w_{s^*}^{(i)} \leftarrow w_{s^*}^{(i)} + 1$

Way-Points and Waiting Times: For each D_i, the ferry will choose a set of cells $\mathbf{s}_{opwp}^{(i)}$ and the corresponding waiting times $\mathbf{w}^{(i)}$. In OPWP, the initial gateway distribution is assumed to be the steady-state distribution whenever the ferry moves to a new way-point (a cell). Based on this assumption, the First Visit Time (FVT) for cell s in D_i, denoted by $\sigma_s^{(i)}$, is defined as the first time the gateway enters cell s (starting from time 0), where its initial distribution is the steady-state distribution. Let $\mathbf{b}_t^{(i,s)}$ be the gateway distribution at t after the ferry waits at cell s for $t-1$ slots without contact, and denote the distribution of FVT by $\Pr_i(s,t) \triangleq \Pr(\sigma_s^{(i)} = t | \mathbf{b}_1^{(i,s)} = \mathbf{b}_0^{(i)})$. Similar to the belief update in π_l^{MY}, we can compute $\Pr_i(s,t)$ by $\Pr_i(s,t) = p_t^{(i,s)} \prod_{m=1}^{t-1}(1 - p_m^{(i,s)})$, where $p_t^{(i,s)} = (b_t^{(i,s)})_s$ and $\mathbf{b}_{t+1}^{(i,s)} = \mathbf{P}_i^T (\mathbf{b}_t^{(i,s)})_{\setminus s}$. The physical meaning of $p_t^{(i,s)}$ is the conditional probability for the gateway to visit cell s at t, assuming no visit before t. According to [8], the objective function of the way-point and waiting time selection in D_i is:

$$\begin{aligned} &\min \sum_{s\in S_i} w_s^{(i)} \\ &\text{subject to } \sum_{s\in S_i} \sum_{t=1}^{w_s^{(i)}} \Pr_i(s,t) \geq p_i, \end{aligned} \quad (21)$$

and all domains have the same p_i.

Lemma VI.1: $\Pr_i(s,t)$ is monotonic, non-increasing with t.

Proof: See Appendix. ■

Lemma VI.1 shows that the probability of contact in cell s at t becomes smaller with increasing t. Therefore, the waiting times can be selected by the greedy algorithm Algorithm 1, and the way-points are simply the cells with positive waiting times. The order of visiting these way-points is arbitrary since it is assumed that the ferry is able to move to any cell in a domain within 1 slot.

Connecting Paths: Since the distances between domains do not necessarily satisfy the *triangle inequality*, the problem of computing inter-domain paths is a general TSP problem, but can nevertheless be solved optimally (by brute force) if the number of domains is small. After the shortest path is determined, the ferry will follow this path to visit each domain and each way-point within it repeatedly. In addition, the best value of p_i can be chosen via simulations (see Fig. 3).

Since OPWP is predetermined, it does not require any on-line calculations. For the global MY, TS, and their approximations, on-line calculations are required; nevertheless,

TABLE I
COMPUTATIONAL COMPLEXITIES OF GLOBAL POLICIES

S-MY	$\Omega\big((\widehat{d}-1)l_{max}\big)$
S-TS	$\Omega\big((\widehat{d}-1)^2 l_{max}^2\big)$
S-TSA^2	$\Omega\big((\widehat{d}-1)^2 l_{max}\big)$
S-TSA^2-N	$\Omega\big(2(\widehat{d}-1)l_{max}\big)$
S-TSA^{1,2}	$\Omega\big((\widehat{d}-1)^2\big)$
OPWP	$O(1)$

using run-time observations can enable dynamic adaptation to serve the gateways more efficiently. A comparison of the computational complexity for all the steady-state-based policies is summarized in Table. I.

VII. SIMULATION RESULTS

In this section, the performance of the proposed hierarchical policies are evaluated and compared with the predetermined policy OPWP.

A. Gateway Mobility Model

Suppose the gateway in D_q follows a 2-D random walk whose transition matrix $\mathbf{P}_q$ is governed by an *activeness parameter* ξ_q and a *tightness parameter* η_q as in [10]. Here ξ_q denotes the likelihood for the gateway to move to other cells: $\xi_q \triangleq \sum_{j\neq i}(P_q)_{ij}$, whereas η_q denotes the bias in its moving directions, with $(P_q)_{ij} \propto e^{-\eta_q ||h-j||_1}$, where h is the home cell (chosen to be the domain center) and $||h-j||_1$ is the taxicab distance between cells h and j.

B. Parameter Settings

Let $\mathbf{L}$ be the traffic demand matrix, where $L_{ij} = (\lambda_i)_{j,l_{max}}$ in (5). Then the settings of $\mathbf{L}$ and the distance matrix $\mathbf{d}$ are

$$\mathbf{L} = \begin{pmatrix} 0 & 0 & 0.5 & 2 & 1 & 1 \\ 1 & 0 & 2 & 2 & 1.5 & 0 \\ 1.5 & 1 & 0 & 3 & 0 & 0.5 \\ 1 & 1 & 0 & 0 & 2 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 3 & 0 & 1.5 & 0 \end{pmatrix},$$
$$\mathbf{d} = \begin{pmatrix} 0 & 3 & 3 & 5 & 1 & 1 \\ 3 & 0 & 5 & 5 & 1 & 2 \\ 3 & 5 & 0 & 4 & 2 & 3 \\ 5 & 5 & 4 & 0 & 2 & 1 \\ 1 & 1 & 2 & 2 & 0 & 4 \\ 1 & 2 & 3 & 1 & 4 & 0 \end{pmatrix}. \tag{22}$$

In addition, $\widehat{d} = 6$, $l_{max} = 90$, $\beta = 0.98$, and the simulation time is 160 slots. For other parameters, we consider both the homogeneous (Fig. 4) and heterogenous (Fig. 5) domain settings to evaluate the proposed policies. Using the same parameter setting, the discounted effective throughput under OPWP is plotted in Fig. 3 for various p_i, which shows OPWP performs best when $p_i = 0.9$ for both homogeneous and heterogenous domain settings. Accordingly, $p_i = 0.9$ will be used to compare OPWP and SAN. Moreover, since $\widehat{d}$ is small, the method of exhaustive enumeration is used to compute the shortest inter-domain path in OPWP.

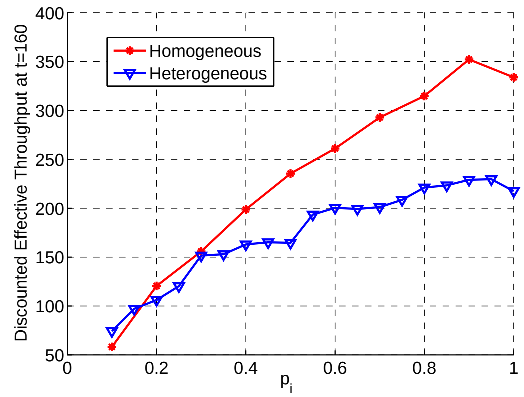


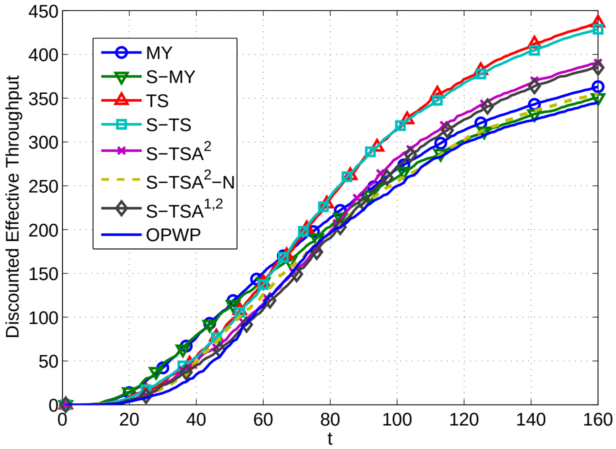
Fig. 3. Best cumulative probability p_i in OPWP

C. Performance Comparison

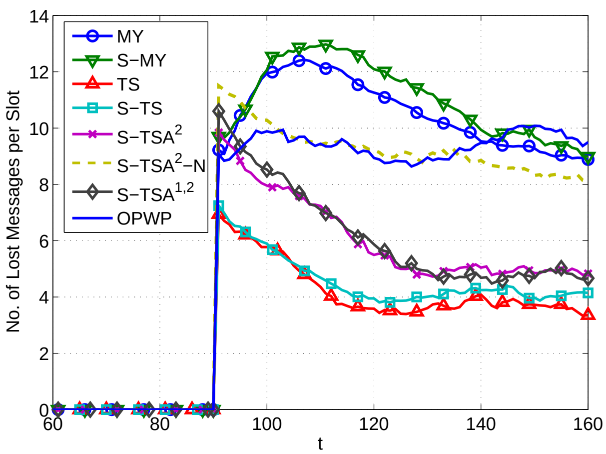
The simulation results are sketched in Fig. 4 and Fig. 5. The results show that the performance of TS is the best, yielding significant improvement over OPWP, whereas MY cannot beat OPWP in the long run due to its shortsighted decisions. S-TS, S-TSA^2 and S-TSA^{1,2} sacrifice some performance gain in exchange for computation efficiency, but they still outperform OPWP. A closer look at Fig. 4(b) and Fig. 5(b) shows that the message loss ratio gradually declines for all SAN policies but not for OPWP. This illustrates the capability of dynamic policies to learn the run-time situation through observations. The steady-state-based approximations S-MY, S-TS resemble the original closely. The further approximations S-TSA^2-N performs similarly to OPWP with a relatively large message loss ratio, whereas S-TSA^2 and S-TSA^{1,2} maintain the performance advantage of TS with much less computation, especially for S-TSA^{1,2} with only $\Omega\big((\widehat{d}-1)^2\big)$ complexity. Although the number of contacts of OPWP is comparable to that of SAN policies (Fig. 4(c) and Fig. 5(c)), its message delivery rate is lower because it neglects the buffer states of the ferry and the gateways. In summary, the dynamic SAN policy outperforms the predetermined OPWP policy with just two-step optimization, even under certain approximations. However, this is in sharp contrast with the results in [10], where myopic policy already outperforms the best predetermined policy, which shows the significance of planning over multiple steps when relaying delay-sensitive messages among scattered domains.

VIII. CONCLUSIONS AND FUTURE WORK

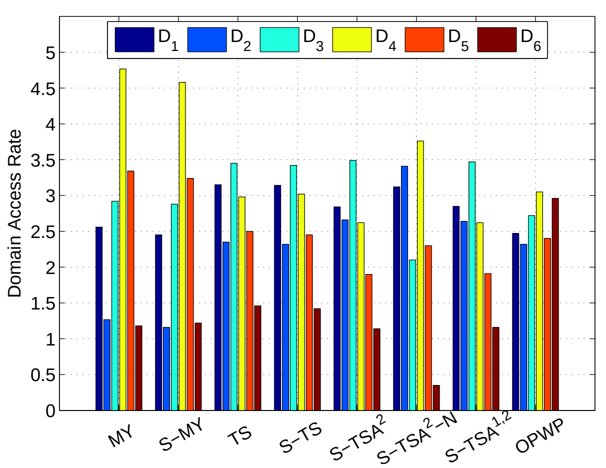
We propose a hierarchical control framework of Switch-and-Navigate (SAN) for controlling a data ferry in a highly partitioned mobile network with bounded delay constraints. In SAN, the global control utilizes the buffer states and the beliefs of gateway locations in all domains to select the best domain to switch to. When the ferry enters a domain, the local control is activated to navigate the ferry for fast searching of the gateway. Simulation results show that SAN outperforms the best known predetermined policy with just one-step lookahead. Moreover, approximation policies are developed for better computational



(a) Discounted effective throughput

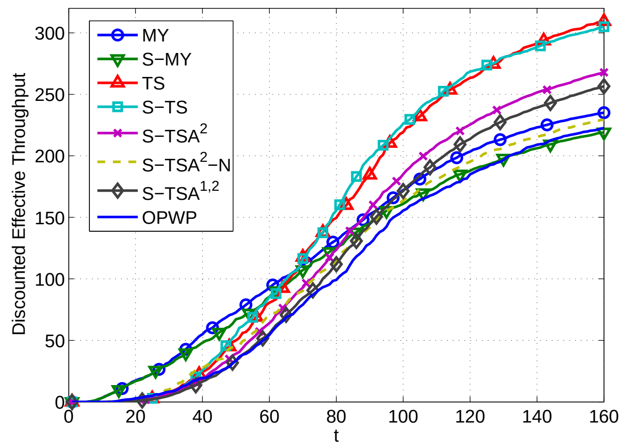


(b) Message loss ratio

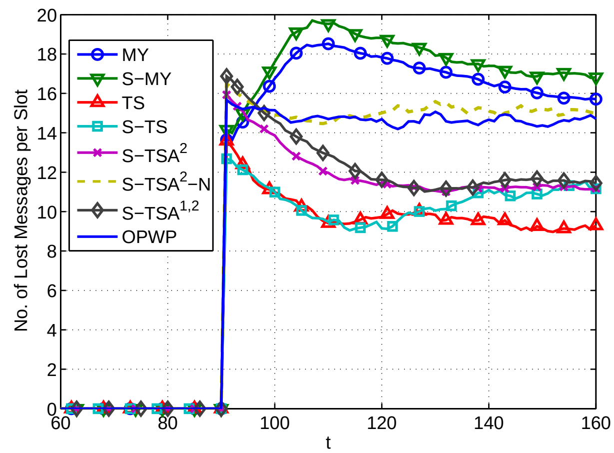


(c) Number of contacts in each domain

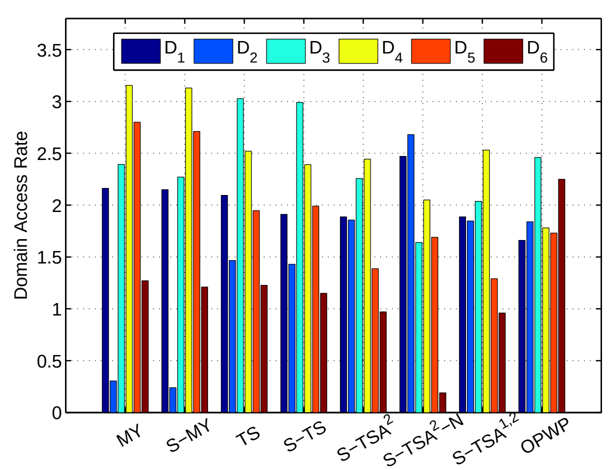
Fig. 4. Homogeneous domain settings: $N_q \equiv 9$, $\xi_q \equiv 0.5$, $\eta_q \equiv 0.03$, $p_q \equiv 0.9$, $q \in \mathfrak{D}$, 240 Monte Carlo runs.



(a) Discounted effective throughput



(b) Message loss ratio



(c) Number of contacts in each domain

Fig. 5. Heterogeneous domain settings: $\mathbf{N} \triangleq (N_q)_{q=1}^{\widehat{d}} = (9, 16, 4, 25, 16, 9)$, $\boldsymbol{\xi} \triangleq (\xi_q)_{q=1}^{\widehat{d}} = (0.7, 0.4, 0.6, 0.8, 0.9, 0.3)$, $\boldsymbol{\eta} \triangleq (\eta_q)_{q=1}^{\widehat{d}} = (0.1, 0.03, 0.05, 0.08, 0.3, 0.2)$, $p_q \equiv 0.9$ ($q \in \mathfrak{D}$), 240 Monte Carlo runs.

efficiency in practical applications. In the future, the scenario of joint control of multiple ferries will be explored.

REFERENCES

[1] J. Leguay, T. Friedman, and V. Conan, "DTN routing in a mobility pattern space," in *ACM SIGCOMM workshop on Delay-tolerant networking*, Philadelphia, USA, Aug. 2005.

[2] A. A. Somasundara, A. Ramamoorthy, and M. B. Srivastava, "Mobile element scheduling for efficient data collection in wireless sensor networks with dynamic deadlines," in *IEEE RTSS*, Lisbon, Portugal, Dec. 2004.

[3] A. Kansal, A. A. Somasundara, D. Jea, M. B. Srivastava, and D. Estrin, "Intelligent fluid infrastructure for embedded networking," in *ACM MobiSys*, Boston, USA, Jun. 2004.

[4] R. C. Shah, S. Roy, S. Jain, and W. Brunette, "Data mules: modeling and analysis of a three-tier architecture for sparse sensor networks," *Ad Hoc Networks*, vol. 1, pp. 215–233, Jan 2003.

[5] A. Chakrabarti, A. Sabharwal, and B. Aazhang, "Using predictable observer mobility for power efficient design of sensor networks," in *IPSN*, Palo Alto, USA, Apr. 2003.

[6] W. Zhao and M. H. Ammar, "Message Ferry: Proactive routing in highly-partitioned wireless ad hoc networks," in *IEEE FTDCS*, San Juan, Puerto Rico, May 2003.

[7] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in *ACM MobiHoc*, Tokyo, Japan, May 2004.

[8] M. M. Bin Tariq, M. Ammar, and E. Zegura, "Message ferry route design for sparse ad hoc networks with mobile nodes," in *ACM MobiHoc*, Florence, Italy, May 2006.

[9] Y. Gu, D. Bozdag, E. Ekici, F. Ozguner, and C.-G. Lee, "Partitioning based mobile element scheduling in wireless sensor networks," in *IEEE SECON*, Santa Clara, USA, Sep. 2005.

[10] T. He, K.-W. Lee, and A. Swami, "Flying in the dark: controlling autonomous data ferries with partial observations," in *ACM MobiHoc*, Chicago, USA, Sep. 2010.

[11] R. D. Smallwood and E. J. Sondik, "The optimal control of partially observable markov processes over a finite horizon," *Operations Research*, vol. 21, no. 5, pp. 1071–1088, Sep.-Oct. 1973.

APPENDIX

Proof of Lemma VI.1: Considering domain D_i, let $\mathbf{z}$ be a zero row vector except $z_s = 1$. For a given cell s in D_i (subscript and superscript i are dropped for simplicity),

$$p_1^{(s)} = \mathbf{z}\mathbf{b}_0,\ p_2^{(s)} = \frac{\mathbf{z}(\mathbf{P}^T\mathbf{R}_s)\mathbf{b}_0}{1-p_1^{(s)}}, \dots,\ p_t^{(s)} = \frac{\mathbf{z}(\mathbf{P}^T\mathbf{R}_s)^{t-1}\mathbf{b}_0}{\prod_{j=1}^{t-1}(1-p_j^{(s)})}. \tag{23}$$

Plug (23) into $\Pr(s,t)$, $\Pr(\sigma(s) = t|\mathbf{b}_1^{(s)} = \mathbf{b}_0) = \mathbf{z}(\mathbf{P}^T\mathbf{R}_s)^{t-1}\mathbf{b}_0$ $(t \geq 1)$. Let column vector $\mathbf{c}_t \triangleq (\mathbf{P}^T\mathbf{R}_s)^{t-1}\mathbf{b}_0$, then $\Pr(s,t) = (c_t)_s$. Since $\mathbf{I}-\mathbf{R}_s$ is a square $\mathbf{0}$ matrix except $(I-R_s)_{s,s} = 1$,

$$\mathbf{c}_1 - \mathbf{c}_2 = \mathbf{b}_0 - \mathbf{P}^T\mathbf{R}_s\mathbf{b}_0 = \mathbf{P}^T(\mathbf{I}-\mathbf{R}_s)\mathbf{b}_0 \succeq \mathbf{0}. \tag{24}$$

Using induction, supposing $\mathbf{c}_{t-1} - \mathbf{c}_t \succeq \mathbf{0}$, then

$$\begin{aligned} \mathbf{c}_t - \mathbf{c}_{t+1} &= (\mathbf{P}^T\mathbf{R}_s)^{t-1}\mathbf{b}_0 - (\mathbf{P}^T\mathbf{R}_s)^t\mathbf{b}_0 \\ &= (\mathbf{P}^T\mathbf{R}_s)(\mathbf{c}_{t-1} - \mathbf{c}_t) \succeq \mathbf{0}. \end{aligned} \tag{25}$$

Thus, $\Pr(s,t) - \Pr(s,t+1) = (c_t)_s - (c_{t+1})_s \geq 0$ for all $t \geq 1$. Consequently, $\Pr(s,t)$ is a monotonic, non-increasing function with t for any given s. ■