Minimizing Wait Time and Age of Information in Mobility-Enabled Communication Systems

Winston Hurst and Yasamin Mostofi

***Abstract*—Recent advances in robotics present new paradigms in communication systems, particularly the use of autonomous vehicles to enhance communication capabilities in a number of settings. Polling systems, in which a single server services multiple queues and incurs some idle time when switching between these, provide useful models for a number of scenarios encountered in such systems. In this paper, we extend traditional polling systems to model these mobility-enabled communication scenarios by embedding the polling systems in 2D Euclidean space and defining service regions in this space where the queues' requests may be serviced. We show that this model can be used to capture the Age of Information (AoI) in the system as well. We pose the fundamental problem of minimizing the average wait time for a request, a key metric of the system's performance, and we provide lower and upper bounds on the optimal solution. Focusing on a cyclic visit order, we find provably optimal trajectories for the low- and high-traffic cases. Finally, we present numerical results from an IoT data collection scenario which illustrate the derived theory and show significant improvements in average wait time when compared to a move-stop-communicate baseline.**

***Index Terms*—mobility-enabled communication, polling systems, optimization, Age of Information**

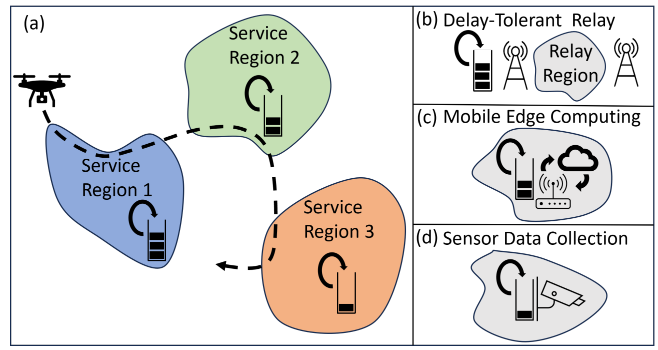

Fig. 1: (a) An example three-queue Euclidean Polling System with Contiguous Service Regions (EPS-CSR). (b-d) Example scenarios modeled by queues with service regions. (b) Data accumulating at a source that must be sent to a destination. (c) An IoT device requires the fulfillment of edge computing tasks. (d) Data gathered by a remote sensor must be offloaded. The service region corresponds to the area where (b) relay between the source and destination is possible or (c, d) data may be transferred from the IoT device or sensor to the AV. See color PDF for better viewing.

I. Introduction

With continued advancement in the capabilities of autonomous vehicles (AVs), wireless communication networks increasingly incorporate these vehicles into system design [1], [2]. These AVs can be used for many tasks, including dynamic coverage in the case of emergency, data collection from IoT devices and wireless sensors, or even mobile edge computing. We refer to the use of AVs in communication networks as *communication-aware robotics* (CAR) [3].

A polling system, in which a single server provides service multiple queues, naturally models many of the CAR scenarios mentioned above. In these cases, the AV acts as the server, while the queues may represent data accumulating at a sensor or computational tasks that must be offloaded. However, while polling systems have long been useful models for many aspects of communication systems [4], their application in CAR settings has received less attention.

To bring the powerful analytical tools of queuing theory and polling systems to bear more fully on CAR problems, this paper presents an extension of traditional polling systems in which the server operates in a 2D Euclidean space and, for each queue, there is a region in which the queue may be served (see Fig. 1). This more completely models many CAR scenarios by allowing the server to work on servicing requests while moving through space, thus reducing switching times. We show how to optimally take advantage of this mobility to minimize the average wait time of the system.

Examples of the application of polling systems for CAR are found in [5], which studies the problem of multiple robots tasked with persistent data collection, and more recently in [6], which studies the problem of minimizing the average wait time in a system of disparate relay links serviced by an AV. There has also been significant interest lately in related problems investigating trajectory design for the minimization of the Age of Information (AoI) [7]–[10]. However, these works assume a move-stop-communicate (MSC) trajectory, in which the AV remains stationary while communicating. There are two important drawbacks to these trajectories. First, they ignore advantages of continued mobility. For example, [11] shows significant energy efficiency gains when using a non-MSC protocol compared to an MSC protocol for a UAV that provides communication to users. Second, MSC trajectories are not practicable for all AVs, *e.g.*, fixed-wing UAVs.

Within polling systems literature, a rich body of work analytically characterizes the behavior of the system for many different scenarios [4]. However, to our knowledge, no work has investigated polling systems as considered here (embedded in Euclidean space and with service regions). Furthermore, optimization of polling systems continues to be an area with many open questions [12]. Compared to existing work, our

Winston Hurst and Yasamin Mostofi are with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, USA (email: {winstonhurst, ymostofi}@ece.ucsb.edu). This work was supported in part by NSF RI award 2008449 and ONR award N00014-23-1-2715.

extension introduces a new degree of freedom when planning the server's operation - its movement through space.

The chief contributions of our work are enumerated below:

- We present an extended polling system model embedded in 2D Euclidean space where requests may be serviced anywhere within a queue's *service region*, here assumed to be contiguous. This model applies to various communication-aware robotics scenarios and beyond, and we also show that it can model Age of Information (AoI). We refer to this model as the *Euclidean polling system with contiguous service regions (EPS-CSR).*
- We pose the general problem of finding optimal policies that minimize the average wait time of the system and show that "best case" and "worst case" polling systems provide lower and upper bounds, respectively, on the performance of the EPS-CSR.
- Focusing on cyclic visit orders, we derive provably optimal solutions for the minimal average wait time problem in low- and high-traffic regimes.
- We provide a detailed numerical example illustrating the application of the model to an IoT data collection scenario. The results corroborate our theoretical findings and show that our trajectories significantly outperform MSC trajectories. Additionally, we explore the impact of energy-related system parameters on system performance.

II. System Model

In this section, we first give a brief introduction to polling systems before presenting our extension in Euclidean space with contiguous service regions.

A. Polling Systems Primer

A polling system (PS) consists of a set $N = \{1, ..., n\}$ queues and a single server. We define the system as a triple $(\mathcal{A}, \beta, \Sigma)$, where $\mathcal{A} = \{\mathcal{A}_i\}_{i\in N}$ is the set of arrival processes at each queue with λ_i denoting the average arrival rate, $\beta = \{\beta_i\}_{i\in N}$ describes service time distributions (*i.e.*, the amount of time required to service a single request) with average $\bar{\beta}_i$, and Σ the set of switching times between queues, with s_{ij} the switching time between the i^{th} and j^{th} queues. The traffic at each queue is then given by $\rho_i = \lambda_i \bar{\beta}_i$, and for the whole system, $\rho_s = \sum_{i=1}^{n} \rho_i$.

In this paper, we model each queue as $M/G/1$, so that $\mathcal{A}$ consists of independent Poisson processes, and the service times are independently and identically distributed according to an arbitrary distribution. We further assume non-preemptive servicing, *i.e.*, the server completely services a request before starting the next. We treat switching times as deterministic, though much of PS literature considers the stochastic case. Relaxing some of these assumptions sets out an interesting direction for future work.

In general, the server may be in one of three states: servicing (actively fulfilling a request), switching (moving from one queue to the next), or idling (sitting at a queue but not servicing). The server's operation may be decomposed into a series of stages which begin at each consecutive *polling instant* (moment when the server begins to service a queue). When at a queue, the server follows a *service policy*, which determines how long to remain at a queue. At each *service completion instant,* it queries a *routing policy*, which determines the sequence, $\mathcal{Q} = \{q_k\}_{k\geq 0}$, $q_k \in N$, of visits to the queues.

B. Polling Systems with Continuous Servicing Regions

We now extend the traditional PS model by embedding it in Euclidean space and allowing queues to be serviced within contiguous service regions. The server moves with bounded velocity, $v \in [0, v_{\max}]$, through a 2D workspace, and it may only service the i^{th} queue when it is within a known, contiguous region $R_i \subset \mathbb{R}^2$. Here, we assume the service rate is constant within the region and leave to future work the case where the rate varies. The only constraint we put on the server's trajectory, $x(t) \in \mathbb{R}^2$, is that it be continuous. This Euclidean polling system with contiguous service regions (EPS-CSR) is described by the triple $(\mathcal{A}, \beta, \{R_i\}_{i\in N})$.

In this model, each stage starts at the polling instant, which corresponds to the moment when the server enters a region R_i[1]. Each stage may be further decomposed into a servicing interval, during which the server moves within the region while either servicing requests or idling, and a switching interval, during which the server moves to the next region. Furthermore, let $x_{k,s}$, $x_{k,c} \in R_{q_k}$, be the robot's position at the k^{th} polling instant and service completion instant, respectively, with corresponding timestamps $t_{k,s}$ and $t_{k,c}$, *i.e.*, $x(t_{k,s}) = x_{k,s}$ and $x(t_{k,c}) = x_{k,c}$ (see Fig. 2).

Let ϕ denote the generalized service policy, which schedules the timing and number of requests serviced and determines the trajectory while the queue is being serviced. Without loss of generality, the server will immediately service any requests found in the queue, *i.e.*, it will not idle when the queue is non-empty, though it may choose to switch before exhausting the queue. The policies may be parameterized by the time until the next service completion instant, τ, the next target service completion point $\tilde{x}_{q,c}$, and a trajectory $\mathcal{T}$ from the current location to the service completion location which is consistent with τ and $\tilde{x}_{q,c}$, so that $\phi(t) = (\tau, \tilde{x}_{q,c}, \mathcal{T})$.

Let μ be the generalized routing policy, which determines not only which queue to visit next, but also how to move between R_{q_k} and $R_{q_{k+1}}$. We assume that the entry point for the next region, $x_{k+1,s}$ is determined and committed to at the time of departure from the previous region, $t_{k,c}$, that is, the server does not dynamically update the next queue to visit or the entry point while moving between the two regions. This reflects the realistic scenario where real-time information on the system state may not be available to the server at all times. Without loss of generality, we consider only switching policies which have the server moves in a straight line between $x_{k,c}$ and $x_{k+1,s}$ with speed $v_{\max}$, so that the policy has the form $\mu(t_{k,c}) = (q_{k+1}, x_{k+1,s})$. The complete trajectory is determined by iteratively following ϕ while servicing and μ while switching to the next region.

[1]This differs slightly from the model of traditional polling systems, in that we treat consecutive visits as a single visit.

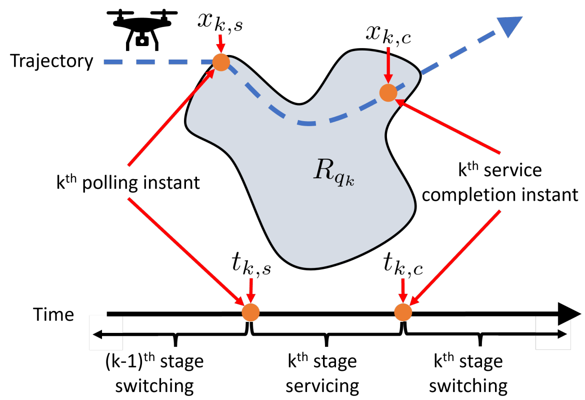

Fig. 2: Stage-based decomposition of a mobile server's trajectory. While servicing, the server follows service policy ϕ, and while switching, it follows switching policy μ.

C. Average Wait Time

A key performance metric of a polling system (PS) is the average wait time, that is, the average duration of time a request waits in a queue before service begins. This has clear relevance to many communication systems, where requests may be taken as, *e.g.*, bits of data accumulating at remote nodes. We assume the server's operation permits a stationary regime, so that Little's theorem gives the average wait time for the i^{th} queue as $\bar{W}_i = \bar{L}_i/\lambda_i$, where is $\bar{L}_i$ the average length of the i^{th} queue within the stationary regime. Assuming ergodicity of the underlying process, this can be reformulated as

$$\bar{W}_i = \lim_{T\to\infty} \frac{1}{T\lambda_i} \int_{t=0}^{T} l_i(t)dt\,. \tag{1}$$

While not discussed in this paper, characterizing choices of μ and ϕ for which the limit in Eq. (1) exists is an interesting question for future work. The long-run proportion of bits entering queue i is given by λ_i/λ_s, with $\lambda_s = \sum_{i=1}^{n} \lambda_i$, so the average wait time across the system is given by

$$\bar{W}_s = \sum_{i\in N} (\lambda_i/\lambda_s)\bar{W}_i. \tag{2}$$

Relationship with Age of Information: Wait time and Age of Information (AoI) both measure the timeliness of data collection, and in fact, long-run average AoI can be thought of as the wait time of a certain PS. Specifically, treating requests as infinitely divisible and assuming a deterministic arrival process, *i.e.*, considering a *fluid polling system* [13], the queue length dynamics are given by

$$\dot{l}_i = \begin{cases} \lambda_i - (1/\bar{\beta}_i) & \text{while being serviced} \\ \lambda_i & \text{otherwise} \end{cases},$$

and to recover the AoI dynamics consistent with the discrete-time dynamics given in [7], [8][2], we let $\lambda_i = 1$ and take the limit as $\bar{\beta}_i \to 0$. We can then express the long-run average AoI as in (1) and (2).

[2]Other definitions of AoI that appear in the literature can be cast as wait time problems, too. For expositional clarity, we have used this one.

III. Minimizing Average Wait Time

The problem of minimizing average wait time of an EPS-CSR S can be succinctly stated as follows:

$$\min_{\mu,\phi} \quad \bar{W}_S \tag{3}$$

with solution $\bar{W}^*_S$. While such a general problem is difficult to solve directly, we next present bounds on the general optimal solution in terms of two polling systems derived from EPS-CSR, defined below:

Definition 1. *For an EPS-CSR $S = (\mathcal{A}, \beta, \{R_i\}_{i\in N})$, the Best Case Polling System (BCPS), $S_w = (\mathcal{A}, \beta, \Sigma_b)$, is a polling system with the switching times given by $s_{i,j} = \min_{x_i\in R_i, x_j\in R_j} ||x_i - x_j||/v_{max}$. Furthermore, $\bar{W}^*_{S_b}$ is the associated minimal achievable average wait time in S_b.*

Definition 2. *For an EPS-CSR $S = (\mathcal{A}, \beta, \{R_i\}_{i\in N})$, the Worst Case Polling System (WCPS), $S_w = (\mathcal{A}, \beta, \Sigma_w)$, is a polling system with the switching times given by $s_{i,j} = \max_{x_i\in R_i, x_j\in R_j} ||x_i - x_j||/v_{max}$. Furthermore, $\bar{W}^*_{S_w}$ is the associated minimal achievable average wait time in S_w.*

Theorem 1. *The optimal wait time of an EPS-CSR, S, is bounded by the optimal wait times of the WCPS, S_w, and BCPS, S_b, that is, $\bar{W}^*_{S_b} \le \bar{W}^*_S \le \bar{W}^*_{S_w}$.*

Proof. We first show that $\bar{W}^*_{S_w}$ is an upper bound by mapping the optimal policy for the WCPS to an analogous policy in the EPS-CSR. First, let $(s_1, s_2, ...)$ be any realization of the sequence of switching times with associated visit order Q in the WCPS under a policy that achieves $\bar{W}^*_{S_b}$. From the definition of the WCPS, we can achieve switching times $(s'_1, s'_2, ...)$ with visit order Q in the EPS-CSR such that $s_i \ge s'_i$, $\forall i$. At each servicing interval, the EPS-CSR server may start servicing the current queue $s_{k-1} - s'_{k-1}$ seconds sooner and can serve the same number of requests (by idling until requests arrive, if necessary), keeping the service completion instants identical across the two systems, so that $l_{i,S}(t) \le l_{i,S_w}(t)$, $\forall t$, $\forall i \in N$, where $l_{i,S}$ and l_{i,S_w} are the length of the i^{th} queue in S and S_w, respectively. Thus, from Eq. (2), $\bar{W}^*_S \le \bar{W}^*_{S_w}$.

As we have guarantees that the switching times in the BCPS are less than or equal to the switching times in the EPS, we can follow a similar argument to show $\bar{W}^*_S \ge \bar{W}^*_{S_b}$. $\square$

While this result provides insight into (3) and the relationship between the EPS-CSR and standard polling system models, a general solution is still out of reach. One possibility is to adapt optimal policies for the associated BCPS or WCPS to the EPS-CSR, an avenue to investigate in future work. For a few important cases, however, optimal solutions can be found, as presented next.

IV. Solutions for Cyclic Policies

In this section, we study the problem of minimizing average wait time for a cyclic visit sequence. We first consider the low traffic system as $\rho \to 0$, which can be used to model the AoI scenario, as discussed in Section II. Then, we consider optimal trajectories as $\rho \to 1$. For both scenarios, we provide a method to find the optimal cyclic visit order.

Without loss of generality, we define the cycle time, t_c, as the time between consecutive visits to the first queue for an arbitrary cycle, and the cycle switching time, t_s, as the total switching time during an arbitrary cycle. We additionally assume that the server follows an exhaustive policy so that it immediately departs for the next region when the current queue is emptied. While this makes the constructed policies in the proof of Theorem 1 invalid (they are allowed to idle at a queue if necessary), we can prove an analogous result for the cyclic, exhaustive scenario.

Lemma 1. *For an EPS-CSR S, the optimal average wait time achievable by a cyclic visit order and exhaustive service policy, $\bar{W}^*_{S,c}$, is bounded by $\bar{W}^*_{S_b,c} \leq \bar{W}^*_{S,c} \leq \bar{W}^*_{S_w,c}$, with $\bar{W}^*_{S_b,c}$ and $\bar{W}^*_{S_w,c}$ the optimal average wait times achievable by a cyclic visit order and exhaustive service policy for the BCPS and WCPS, respectively.*

Proof. Following [14], average wait time is monotonically increasing in cycle switching time, t_s, and clearly, the optimal cycle switching time of the EPS-CSR will always be at least the optimal cycle switching time of the BCPS and at most the optimal cycle switching time of the WCPS. □

A. Low Traffic Regime

Low traffic regimes occur when service rates are much higher than the arrival rates, which may be due to service rates being very high or arrival rates very low, with the former corresponding to the AoI regime (see Section II). In these scenarios, we have the following result.

Theorem 2. *In the limit as $\rho \to 0$, the optimal cyclic visit sequence and server trajectory are given by the solution to a traveling salesperson with neighborhoods problem (TSPN) with the regions $\{R_i\}_{i \in N}$ as the neighborhoods.*

Proof. Using properties of polling systems, it can be shown that as $\rho \to 0$, the probability that $t_c = t_s$ converges to 1, and that the average wait time is given by $t_s/2$ [14]. Thus, the solution to the TSPN minimizes the average switching time, which minimizes the average wait time. □

Intuitively, when the traffic is very low, the server cannot take advantage of servicing time to move within the region to reduce switching time, either because the queues are nearly always empty (low arrival rates) or because the data in the queues is serviced nearly instantaneously (high service rates).

This result is consistent with related results derived for AoI, which show that finding the shortest visit cycle or Hamiltonian path minimizes the AoI [7], [8], though in contrast to our result, those assume the sites must be visited exactly and consider one-time rather than persistent data collection.

B. High Traffic Regimes

In this section we study the problem of minimizing average wait time in high-traffic regimes. Our key insight is that when the server spends a non-trivial amount of time servicing a queue, it may continue moving through the service region while servicing, allowing it to significantly reduce switching times. In this light, we present a greedy stage-level policy (GSLP) and show that in high-traffic regimes, it is optimal.

At a high level, our approach seeks to minimize the switching time of the next transition. First, let Q_{TSP} be the cyclic visit order found by solving the symmetric, non-Euclidean traveling salesperson problem (TSP) with n nodes and switching times given by the best case switching times, Σ_b, and consider the following optimization problem to be solved when the robot is at a location $x(t) \in R_{q_k}$ while servicing q_k:

$$\begin{aligned} \min_{x_{k,c},\, x_{k+1,s}} \quad & ||x_{k,c} - x_{k+1,s}|| && \text{(4a)} \\ \text{s.t.} \quad & x_{k,c} \in R_{q_k} && \text{(4b)} \\ & x_{k+1,s} \in R_{q_{k+1}} && \text{(4c)} \\ & d_{ir,k}(x(t), x_{k,s}) \leq \frac{l_{q_k}(t)}{1-\rho_{q_k}} \bar{\beta}_{q_k} v_{\max} && \text{(4d)} \end{aligned}$$

where $d_{ir}(\cdot,\cdot)$ is the minimum intra-region distance between any two points in R_{q_k}. The term $\frac{l_{q_k}(t)}{1-\rho_{q_k}}\bar{\beta}_{q_k}$ is an intra-region time budget, giving the expected time the server has to move within R_k before it exhausts the queue. Problem (4) finds the next service completion and entry points which minimize the switching distance while constraining the service completion point to be reachable within the time budget. Let $\tilde{x}_{k,c}(t)^*$ and $\tilde{x}_{k+1,s}(t)^*$ be the minimizing arguments of (4) when solved at time t. The service and switching policies of the GSLP are then characterized as follows:

$$\begin{aligned} \phi_g(t) &= \left(\tilde{x}^*_{k,c}(t), \frac{l_{q_k}(t)}{1-\rho_{q_k}} \bar{\beta}_{q_k}, \mathcal{T}^* \right) \\ \mu_g(t_{k,s}) &= (q_{\text{TSP}(k+1)}, \tilde{x}^*_{k+1,s}(t_{k,c})) \end{aligned} \quad (5)$$

where $q_{\text{TSP}(k+1)}$ indicates the $(k+1)^{\text{th}}$ queue serviced under visit order Q_{TSP} and $\mathcal{T}^*$ is any trajectory consistent with $\tilde{x}^*_{k,c}(t)$ and $\frac{l_{q_k}(t)}{1-\rho_{q_k}}\bar{\beta}_{q_k}$, and at least one such trajectory is

Algorithm 1: Server Trajectory Under the Greedy Stage-Level Policy (GSLP)

Input: Cyclic visit order Q_{TSP}, Service regions $\{R_i\}_{i \in N}$ **while** *continuing to operate* **do**
  `// Intra-Stage Policy` ϕ_g
  while *current queue is not empty* **do**
    Step 1: Find $x^*_{k,c}(t)$ by solving (4) for budget $\bar{\beta}_{q_k} l_{q_k}(t)/(1-\rho_k)$ from current location $x(t)$.
    Step 2: Find a trajectory $\mathcal{T}^*$ consistent with current service complete location $x^*_{k,c}(t)$ and budget $\bar{\beta}_{q_k} l_{q_k}(t)/(1-\rho_k)$, *e.g.*, the shortest distance path within R_k form $x(t)$ to $x^*_{k,c}(t)$.
    Step 3: Follow $\mathcal{T}^*$ until service on the current request is complete.
  end
  `// Inter-Stage Policy` μ_g
  Step 4: Find $x_{k+1,s}$ by solving (4) given zero budget, which constrains $x^*_{k,c}(t_{k,c}) = x(t_{k,c})$.
  Step 5: Move in a straight line at maximum velocity from $x_{k,c}$ to $x_{k+1,s}$ to begin servicing the next queue.
end

guaranteed to exist by constraint (4d). When R_{q_k} and $R_{q_{k+1}}$ are convex, the problem in (4) is convex and can be solved very quickly on the fly. Otherwise, road-map type methods [15] may be used to quickly find approximate solutions. In practice, the GSLP is characterized by Algorithm 1.

We now show that this policy minimizes average wait times. Conceptually, the proof shows that under high enough traffic, there is almost surely sufficient time budget to move between any two points in the region, thus allowing the EPS-CSR to achieve the best case switching times of the BCPS. To make the proof rigorous, we make the following assumption:

Assumption 1. *The cycle switching time of BCPS under visit order Q_{TSP}, t_{BCPS}, is non-zero, that is, $t_{BCPS} > 0$.*

Theorem 3. *For any EPS-CSR the GSLP (μ_g, ϕ_g) minimizes average wait time in the limit as $\rho_s \to 1$.*

Proof. Due to space constraints, we prove this for deterministic services times $\bar{\beta}_i$. First note that for the BCPS, the average wait time under a cyclic policy is monotonically decreasing with the cycle switching time [16], so that the cyclic visit order given by the TSP minimizes the wait time in the BCPS. If $l_{q_k}(t_{k,s}) \geq l_{q,\text{th}} = \max_{x,x' \in R_{q_k}} d_{ir,q_k}(x, x')/(\bar{\beta}_{q_k} v_{\max})$, the server has enough budget to move anywhere within the region, and in particular constraint (4d) can be ignored. If this holds at every polling instant, then the server always achieves the switching times of the BCPS, and Lemma 1 can be applied to show optimality. We now prove that we can make $\Pr\big(l_{q_k}(t_{k,s}) < l_{q,\text{th}}\big)$ arbitrarily small as $\rho_s \to 1$.

Let the random variable T_i denote the elapsed time between a polling instant and the previous service completion instant for the i^{th} queue when following the GSLP. The total amount of data in queue i at a polling instant is distributed as a Poisson random variable, $D_i \sim \text{Poisson}(\lambda_i T_i)$. We have $\text{Var}(D_i) = \mathbb{E}[D_i] = \lambda_i \mathbb{E}[T_i]$. Using properties of polling systems, it can be shown that $\mathbb{E}[D_i] \geq (1-\rho_i) t_{\text{BCPS},s}/(1-\rho_s)$. Without loss of generality, we may say that $\lim_{\rho_\to 1} \rho_i < 1$, so that under Assumption 1, $\mathbb{E}[D_i]$ grows unbounded as $\rho_s \to 1$. For an appropriate choice of $\rho \in (0,1)$, we have $\mathbb{E}[D_i] > L_{i,\text{th}}$. Then, using Chebyshev's inequality, we have

$$P(D_i < L_{i,\text{th}}) \leq P(|\mathbb{E}[D_i] - D_i| \geq \mathbb{E}[D_i] - l_{q_k,\text{th}}) \leq 1/\kappa^2$$

where $\kappa = (\mathbb{E}[D_i] - l_{q_k,\text{th}})/\sqrt{\mathbb{E}[D_i]}$, and $\lim_{\rho_s \to 1} 1/\kappa^2 = 0$. Thus, as $\rho \to 1$, we have $P(D_i < l_{q_k,\text{th}}) \to 0$. □

V. Numerical Results

In this section, we illustrate how the EPS-CSR model can be applied to real-world communication-aware robotics problems. We first discuss relevant parameters of the underlying communication system and then relate these back to an EPS-CSR. Numerical results corroborate the theorems presented above, and we explore the impact of energy-related parameters (transmit power and velocity) on system performance.

Consider a UAV flying at a fixed altitude of $h = 400$ m with a max velocity is 10 m/s while assigned to persistently collect data from a set of ten IoT devices placed in the environment as shown in Figure 3 (a). Data accumulates at each queue with a rate of λ_i, and the percent of arrivals at each queue, *i.e.*, $\pi_i := \lambda_i/\lambda_s$, is shown in Figure 3 (a). Given the UAV's height, we assume a dominant line-of-sight channel exists, so that the magnitude squared of the complex channel coefficient between the UAV an IoT device is given by $H = K_o d^{-\alpha}$, where the path loss coefficient K_o is the path loss at a reference distance of 1 m, d is the distance between the UAV and the IoT device, and α is the path loss exponent. Here, we take K_o and α to be $-30dB$ and 2.95, respectively. We further assume the IoT devices employ a 16-QAM modulation scheme and transmit with a maximum power of 20 dBm over a 1 MHz channel. The noise at the UAV receiver is taken to be -75 dBm so that to achieve a bit error rate of at least $1e-6$, the robot must be within 119 m of the IoT device's projection onto the horizontal plane in which the UAV flies [17]. Translating this into an EPS-CSR, we have deterministic service times of $\bar{\beta}_i =$ 16 Mb/s for each queue, and the regions are given by $R_i = \{x \in \mathbb{R}^2 | \, \|x - x_i\|_2 \leq 119\}$, where x_i the is the location of the i^{th} IoT node projected into the flight plane. We next examine the system under various traffic loads.

A. Impact of Traffic on System Performance

In this section, we explore the impact of the traffic at the IoT queues. In doing so, we compare the wait time under four cyclic policies: (1) a baseline move-stop-communicate (MSC) policy based on [**?**], [7], which visits each site exactly with the visit order determined by the corresponding traveling salesperson problem (TSP), (2) a low-traffic optimal (LTO) MSC policy which is found by solving the associated TSPN problem (see Theorem 2), (3) the greedy policy described in Algorithm 1 (GSLP), and (4) the performance of the associated BCPS with the optimal visit order Q_{TSP}, which provides a lower bound on the wait time (see Lemma 1). Figure 3 (a) shows the MSC and LTO trajectories as well as the trajectory of the GSLP in the high traffic regime, which achieves the same performance as the BCPS (see Fig. 3 (b)).

The average wait time for these four systems is simulated for various traffic loads, and the results are plotted in Figure 3 (b). The results are consistent with the theory developed in this paper. For very low traffic loads, i.e., in regimes consistent with an AoI interpretation of the average wait time, the LTO policy outperforms the MSC baseline and GSLP policy, while the BCPS provides an unobtainable lower bound. As the traffic increases, the freedom of movement allowed by the GSLP leads it to outperform the LTO policy, and the system performance converges to the BCPS lower bound. In all traffic conditions, exactly visiting each site results in the worst average wait time.

B. System Performance and Energy Consumption

In the considered scenario, the UAV's velocity and the transmit (TX) power of the IoT devices are two key drivers of system energy consumption. Figure 3 (c) show the results of a study on the impact of both TX power and velocity on average wait time of the system by looking at the BCPS lower bound in a high traffic regime, $\rho_s = 0.8$. As expected, we see that as either velocity or TX power increases, the average wait time

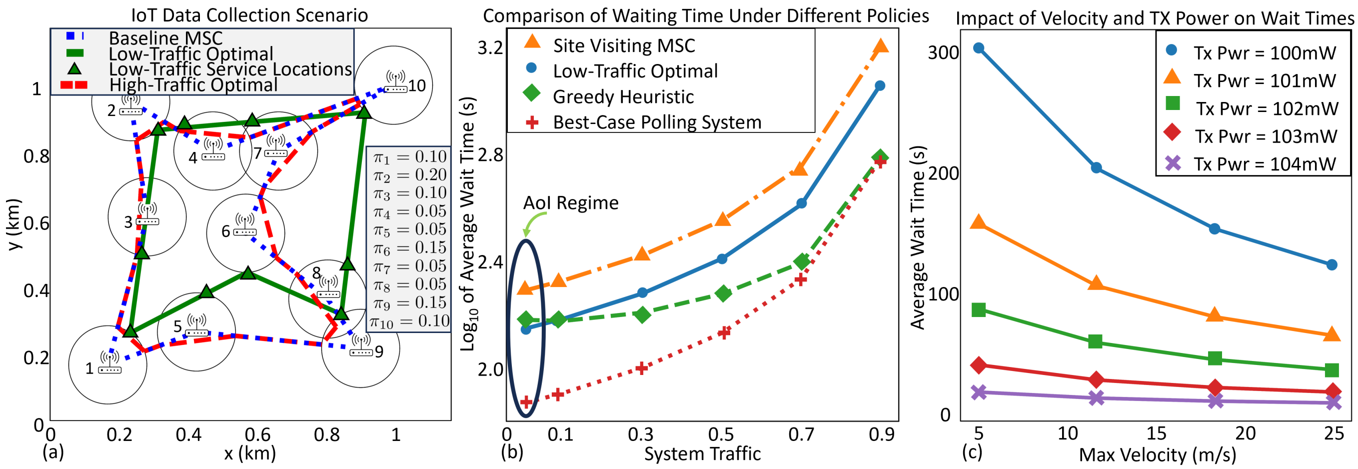

Fig. 3: Results for the IoT data collection numerical example. (a) The high-traffic optimal (HTO) (red), low-traffic optimal (LTO) (green), and baseline most-stop-communicate (MSC) (blue) cyclic trajectories with labeled arrival ratios $\pi_i = \lambda_i/\lambda_s$ and communication regions indicated by black circles. For both the LTO and baseline MSC trajectories, the UAV and IoT devices do not communicate while the UAV moves, but under the HTO policy, communication occurs as the UAV moves through the region. For the high-traffic policy, the straight-line trajectory within the regions is not uniquely optimal. (b) Comparison of average wait time under the baseline MSC policy (orange), the LTO policy (blue), the greedy stage-level policy (GSLP) from Algorithm 9 (green) and the BCPS (red). In the high traffic regime, the performance of the GSLP converges to that of the BCPS. (c) Comparison of average wait time in the BCPS for $\rho_s = 0.8$ with different maximum velocities and transmit powers. Increasing either reduces wait times, but increasing transmit power does so more quickly.

decreases. Moving faster directly reduces the switching time between regions while also giving the UV a greater budget to move within the region, thus also reducing the switching distances themselves. Increasing TX power grows the size of the service region, which also reduces switching times. The study indicates that increasing TX power is more impactful than increasing speed. In particular, note that with 100 mw of TX power, increasing the velocity up to 25 m/s brings the wait time down to 119 s, which is over 30 s more than the wait time achieved when the velocity is 5 m/s but the TX power is increased only slightly to 102 mW. Furthermore, at 104 mW, the regions R_i have grown such that they all nearly overlap, so that the wait time of the system approaches that of an equivalent $M/G/1$ queue (near 0).

VI. Conclusions

In this paper, we have presented a general polling system model capable of capturing the salient features of many communication-aware robotics scenarios, such as data gathering or delay-tolerant relaying. We showed the model was generic enough to additionally capture the concept of AoI, and posed the general problem of average wait time minimization. We then bounded the minimal achievable wait time by that of two related polling systems before finding optimal cyclic trajectories for low- and high-traffic regimes. Finally, we provided a detailed numerical example grounded in an IoT data collection scenario which corroborated our theoretical results, demonstrated the performance gains possible when allowing the server to continue moving while servicing, and explored the impact of energy-related system parameters.

References

[1] 3GPP, "Technical Specification Group Radio Access Network; Study on 3D channel model for LTE (Release 12)," Study TR 36.777, 2018.

[2] Qualcomm, "LTE Unmanned Aircraft Systems: Trial Report (v1.0.1)," Study V 1.0.1, 2017.

[3] A. Muralidharan and Y. Mostofi, "Communication-aware robotics: Exploiting motion for communication," *Annual Reviews of Control, Robotics, and Autonomous Systems*, 2021.

[4] V. Vishnevsky and O. Semenova, "Polling Systems and Their Application to Telecommunication Networks," *Mathematics*, vol. 9, no. 2, 2021.

[5] W. Saad, Z. Han, T. Basar, M. Debbah, and A. Hjorungnes, "A Selfish Approach to Coalition Formation Among Unmanned Air Vehicles in Wireless Networks," in *2009 Int. Conf. on Game Theory for Networks*, 2009, pp. 259–267.

[6] W. Hurst and Y. Mostofi, "Optimization of Mobile Robotic Relay Operation for Minimal Average Wait Time," *IEEE Trans. on Wireless Commun.*, 2022.

[7] Y. Luo, J. Xu, J. Chen, and J. Huang, "UAV Trajectory Planning with Network Age of Information Minimization," in *2022 IEEE Wireless Commun. and Networking Conf. (WCNC)*, 2022, pp. 1862–1867.

[8] J. Liu, X. Wang, B. Bai, and H. Dai, "Age-Optimal Trajectory Planning for UAV-Assisted Data Collection," in *IEEE Conf. on Computer Commun. Workshops (INFOCOM WKSHPS)*, 2018, pp. 553–558.

[9] G. Chen, C. Cheng, X. Xu, and Y. Zeng, "Minimizing the Age of Information for Data Collection by Cellular-Connected UAV," *IEEE Trans. on Vehicular Technology*, vol. 72, no. 7, pp. 9631–9635, 2023.

[10] W. Jiang, C. Shen, and B. Ai, "UAV-Assisted Sensing and Communication Design for Average Peak Age-of-Information Minimization," in *2022 IEEE Int. Conf. on Commun. Workshops (ICC Workshops)*, 2022, pp. 1005–1010.

[11] Y. Zeng, J. Xu, and R. Zhang, "Energy Minimization for Wireless Communication With Rotary-Wing UAV," *IEEE Trans. on Wireless Commun.*, vol. 18, no. 4, pp. 2329–2345, 2019.

[12] V. Vishnevsky and A. V. Gorbunova, "Application of Machine Learning Methods to Solving Problems of Queuing Theory," in *Information Technologies and Mathematical Modelling. Queueing Theory and Applications*, A. Dudin, A. Nazarov, and A. Moiseev, Eds. Cham: Springer International Publishing, 2022, pp. 304–316.

[13] O. Czerniak and U. Yechiali, "Fluid polling systems," *Queueing Systems*, vol. 63, pp. 401–435, 2009.

[14] O. Boxma and J. Weststrate, "Waiting times in polling systems with markovian server routing," in *Messung, Modellierung und Bewertung von Rechensystemen und Netzen. Informatik-Fachberichte*, 1989.

[15] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, 2011.

[16] O. Boxma, "Workloads and Waiting Times in Single-Server Systems with Multiple Customer Classes," *Queueing Systems*, pp. 185–214, 1989.

[17] A. J. Goldsmith, "Wireless Communications." Cambridge Univ. Press, 2005.