Sneakernet and Station Wagons to Robots: Bounds for Robotic Network Throughput*

Matthew A. Schack[1], John G. Rogers[2], and Neil T. Dantam[1]

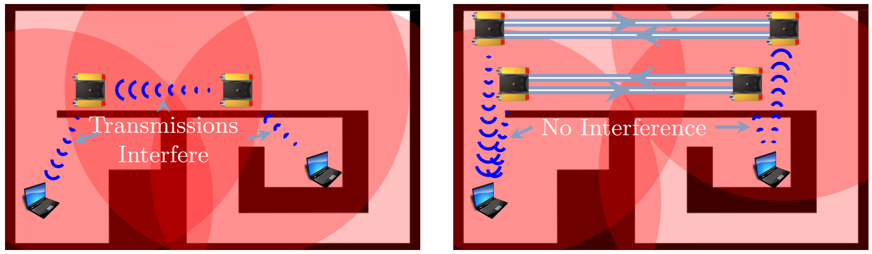
[1] Department of Computer Science, Colorado School of Mines, USA
{mschack,ndantam}@mines.edu
[2] DEVCOM Army Research Laboratory john.g.rogers59.civ@army.mil

Abstract. Robot teams are an effective tool for communicating data in unstructured environments (e.g., data collection). Since robot teams can rearrange themselves, they can ferry data through physical motion rather than only wireless transmissions, potentially increasing network throughput. We construct a Robot Network Graph that represents throughput due to both physical motion and wireless communication, and use the Robot Network Graph to find a polynomial time upper bound on network throughput. We combine the upper bound with an optimization formulation for maximizing network throughput, producing a bounded-optimal solution. We analyze the performance of our bounded-optimal solution and show our bound is at least 3x tighter than a bound derived from the maximum flow across a graph for large (> 20 robots) teams in our tested scenarios.

1 Introduction

Multi-robot teams offer capabilities to explore, inspect, and collect data in unstructured environments, often using wireless communication to relay data between robots and ultimately to an operator (e.g., using a robot team to map a mine and deliver it to surveyors). However, wireless networks must contend with interference between agents, where one robot's transmissions serve as noise for others. Though networking technologies have many strategies to address interference [8,14], the effective noise still limits network throughput. Using mobile robots as network relays offers a singular advantage to address interference—robot motion is controllable. While the communication advantages of mobility are long-known (e.g., the classic quip to "Never underestimate the bandwidth of a station wagon full of tapes." [28]), to the best of our knowledge, formal analysis of wireless throughput limits incorporating controlled mobility has not been previously investigated. We observe in particular that robots physically ferrying data do not cause or suffer from wireless interference and thus can avoid interference via physical rearrangement (see Fig. 1). We analyze the effect of robot motion on network throughput to (1) prove the relationship between network throughput and robot data capacity and (2) inform an effective planning approach for robot networks.

* This work was supported in part by the ARL DCIST CRA [W911NF-17-2-0181] and the NSF [CNS-1823245]
Approved for public release; distribution unlimited.



(a) A Static Network (b) A Dynamic Network

Fig. 1: A static and dynamic network. The robots in the static network must be close enough to communicate, meaning they are also close enough to interfere with other transmissions (shown by the red circles). Conversely, robots drive in the dynamic network, allowing them to avoid interfering with one another.

We analyze how robot motion affects network throughput by framing motion as network throughput. Network throughput captures the rate of data movement between two positions regardless of the method of motion—e.g., wired or wireless communication. Therefore, physically ferrying data has a throughput dependent on the amount of data ferried and the time it takes to ferry data. A robot's motion throughput would then increase with data capacity, meaning that with a high enough data capacity the network throughput is limited by robot-to-robot communication rather than any robot motion.

We find an upper bound for the Robot Network Throughput (RNT) problem by creating a graph with edges that describe throughput bounds due to either motion or communication. We then bound throughput by finding multiple bottleneck paths [3]—i.e., finding paths that maximize the minimum edge weight. Finding bottleneck paths can be done in linear time [3], meaning our bound is quick to compute. Moreover, the upper bound describes the motion for each robot, allowing us to use the upper bound to construct a network throughput lower bound where each robot follows the same path and maximizes the wireless network throughput considering interference.

We find optimality bounds for the throughput through robot networks, considering both controllable robot motion and wireless communication. We discuss related work maximizing throughput with uncontrollable motion, robot data collection, and the physics of interference in Sec. 2 and formally define the Robot Network Throughput (RNT) problem in Sec. 3. We describe an infinite optimization formulation to solve RNT, find performance bounds, and relax the model to a quadratically constrained one (Sec. 4). Using our insights from our formulations, we propose a polynomial time upper-bound Sec. 5, which combined with the relaxed formulation results in a bounded-optimal network throughput. We compare our approaches in different domains in Sec. 6.

2 Related Work

We review related work in network throughput, data collection, and the physics of wireless radio interference.

2.1 Network Throughput

Maximizing throughput through a graph without interference is the classic max-flow problem [6,7,10], and finding a single path that maximizes flow is the max-imin or bottleneck path problem [3]. Both bottleneck path and max-flow problems have linear [3] or approaching linear [4] time complexity. Conversely, considering interference between nodes causes the problem to become NP-hard [8]. Leading approaches have constructed interference and network graphs [14] or optimization models [8] to contend with interference. We solve multiple bottleneck path problems to compute a network throughput upper bound, finding a polynomial time bound on throughput and combine the upper bound with optimization to produce a bounded-optimal solution.

Delay Tolerant Networking (DTN) [15,26] considers how to route data through a network with a changing topology. Such works either consider how the topology is changing to be known [13,14] or uncertain [1,16]. While some works do address using controllable nodes [30]—i.e., robots, such works mention that DTN with controllable motion is primarily a robotics problem. Robotics works that consider robot-to-robot communication primarily focus on minimizing collection time [5,12,22,23]—i.e., the data collection problem, rather than networking metrics. Our work bounds the throughput of DTNs with controllable robotic nodes, combining insights from both robotics and networking communities.

2.2 Data Collection

Works combining robot motion and networking focus on reducing latency through communication with teammates. Leading approaches either pre-plan rendezvous points [21,22,24] or synthesize plans with periodic connectivity guarantees [17,29]. Furthermore, many works view the problem as an extension of the Traveling Salesman Problem [2,5,12,23] and minimize or bound latency on the transmitted data. We view the problem as a flow problem and minimize the data transmission rate (i.e., throughput), rather than ensuring we obtain data quickly.

2.3 Physics of Interference

The maximum information rate (i.e., Channel Capacity) across any medium is bounded by the Shannon-Hartley theorem [27],

$$C = B \log_2 \left(1 + \frac{S}{N}\right) \ , \tag{1}$$

where C is the channel capacity, B is the channel's bandwidth, and S and N are the received signal and noise respectively in `watts`. Equation (1) gives an upper bound on throughput, and typical Wi-Fi performance is necessarily worse than this upper bound. However (1) does capture the relationship between throughput, received signal, and interference. Works that consider interference [8] based on physics consider transmissions to be possible so long as noise is below some threshold. We leave our model general to the specific interference model used but give bounds for interference models where a robot's transmission interferes with its data reception.

3 Problem Definition

We consider the Robot Network Throughput (RNT) problem, where a team of robots must collect and transmit information from environmental sources to static sinks. The robots can move, and their motion changes the network topology. We assume that the robots may pick up and deliver data to any source and sink and that there is always data to pick up. We wish to find a cyclic path and transmission schedule that maximizes the steady-state throughput through the network.

Definition 1. *A Robot Network Throughput (RNT) problem is the tuple* $\Sigma = (\mathcal{X}, \mathcal{R}, \mathcal{S}, \mathcal{T}, \mathcal{C}, d_{\mathtt{max}})$ *where,*

- $\mathcal{X}$ *is the set of all positions—i.e.,* $\mathcal{SE}(2)$ *or* $\mathcal{SE}(3)$*. We separate* $\mathcal{X}$ *into valid and invalid space* $\mathcal{X}_{\mathtt{valid}}$ *and* $\mathcal{X}_{\mathtt{invalid}}$*.*
- $\mathcal{R}$ *is a finite set of robots with time-varying position* $x_r \in \mathcal{X}\ \forall r \in \mathcal{R}$
- $\mathcal{S}$ *is a finite set of sources. The static position of each source is* $x_i \in \mathcal{X}\ \forall i \in \mathcal{S}$
- $\mathcal{T}$ *is a finite set of sinks. The static position of each sink is* $x_i \in \mathcal{X}\ \forall i \in \mathcal{T}$
- $\mathcal{C} : \mathcal{X} \times \mathcal{X} \to \Re^+$ *is a goodput function, which given two positions finds the maximum throughput between them—i.e., maximum throughput without any interference.*
- $d_{\mathtt{max}} \in \Re^+$ *is a data capacity—i.e., how much data each robot can store.*

Goodput measures the number of useful bits delivered over a network per unit time [20]. In our case, goodput measures the maximum throughput without any interference and cannot exceed total throughput. Goodput thus abstracts away low-level networking details to account for reduced throughput due to sending information beyond application data such as packet headers and retransmission.

Similar to other works that have robots ferry data [12,22], we assume that the environment has enough clearance such that each robot can always avoid every teammate, allowing us to plan paths for each robot individually rather than a combined team.

For simplicity, we define the set of all robots, sources, and sinks—i.e., all agents— as $\mathcal{A} \triangleq \mathcal{R} \cup \mathcal{S} \cup \mathcal{T}$. Additionally, we define a Forall Distinct operator, $\forall\mathcal{D}\ r, r' \in \mathcal{A} \triangleq \forall r, r' \in \mathcal{A}, r \neq r'$.

The goal of RNT is to maximize the steady-state throughput at the sink nodes. Since we are concerned with steady-state behavior, we constrain the paths to be cyclic and thus repeatable. The average throughput between two agents is $v_{rr'}, r, r' \in \mathcal{A}$, the starting data for any robot $r \in \mathcal{R}$ is d_{r0}, and the time horizon is $t_{\mathtt{max}}$. A solution to RNT is a set of cyclic paths $\sigma : [0, t_{\mathtt{max}}] \mapsto \mathcal{X}$ and a transmission schedule—i.e., a function that returns the throughput between agents at any point in time—$\mathcal{V} : [0, t_{\mathtt{max}}] \mapsto \Re^{+}$, that maximizes the sum of the average throughput into every sink node.

$$\max_{\sigma,\mathcal{V},t_{\mathtt{max}},v} \quad \sum_{r\in\mathcal{R}\cup\mathcal{S},r'\in\mathcal{T}} v_{rr'} \tag{2}$$

$$\begin{aligned}
\mathtt{s.t.NetworkFlow}\left(\mathcal{V}_{rr'}\right) & \qquad \forall\mathcal{D}\ r,r'\in\mathcal{A} & (3)\\
\mathtt{RobotDynamics}\left(\sigma_r\right) & \qquad \forall r\in\mathcal{R} & (4)\\
\sigma_r\left(0\right)=\sigma_r\left(t_{\mathtt{max}}\right),\sigma_r\left(t\right)\in\mathcal{X}_{\mathtt{valid}} & \qquad \forall t,r\in\mathcal{R} & (5)\\
0\le d_{r0}+\sum_{r'\in\mathcal{A}}\int_{t=0}^{t'}\mathcal{V}_{r'r}\left(t\right)-\mathcal{V}_{rr'}\left(t\right)dt\le d_{\mathtt{max}} & \qquad \forall t',r\in\mathcal{R} & (6)
\end{aligned}$$

Where (3) ensures that all transmissions are physically possible and result in data arriving at a sink node, (4) ensures that all motions across the trajectory are physically possible [11,19,25], and (6) ensures all robots have data if they are transmitting and never receive more than the data capacity.

The RNT problem is NP-hard for static agents [8], meaning it must be at least NP-hard for dynamic robots.

We defined RNT as maximizing throughput rather than other queuing theoretic metrics since networking works addressing interference typically take the view of maximizing throughput [8,14]; however, throughput is not the only metric by which we can evaluate network performance. Other works focused on ferrying data have evaluated performance by optimizing [22] or constraining [23] latency. Since latency only requires the network to transmit an arbitrarily small packet of data, the resulting throughput for a minimum latency answer to RNT may be infinitely close to zero. Similarly, we will show that an optimal throughput answer may involve infinite latency (see Sec. 4.1), implying that we cannot optimize for both throughput and latency in all cases. We show performance trade-offs between latency and throughput in our experiments (see Sec. 6) and discuss bounding latency while maximizing throughput.

4 Semi-Infinite Program and Mixed Integer Relaxation

We construct a semi-infinite program (SIP) to solve RNT (2) by finding trajectories and transmission schedules for every robot to maximize network throughput. Directly solving the SIP requires satisfying an infinite number of constraints. Instead, we use the SIP to analyze throughput bounds of RNT to create a Mixed Integer Quadratically Constrained Program (MIQCP) that is throughput-optimal under infinite data capacity, simplifying the problem to a finite number of constraints. Our relaxation has robots either move or communicate, yielding a simpler model than the original SIP. We discuss an upper bound on network throughput in Sec. 5, allowing us to produce a bounded optimal solution by comparing the results from our relaxation and upper bound.

Sec. 4.1 presents our formulation and uses it to prove throughput bounds for infinite data capacity robots (Theorem 1). Sec. 4.2 describes the relaxation using the throughput bounds and shows that the relaxed model is optimal under infinite data capacity (Theorem 2).

4.1 Semi-Infinite Program

Reasoning about network flows in (3) requires us to reason about the instantaneous throughput at every point along each robot's trajectory, which is an infinite amount of points. Therefore, our formulation is semi-infinite due to having an infinite number of variables, which means the formulation is computationally challenging to solve as we must find both the optimal answer and the set of constraints that bound the optimal answer. We analyze our formulation and find that the trajectory portion where throughput is constant bounds throughput, meaning we can relax our formulation to a quadratically constrained one.

The goodput function, $\mathcal{C}$, defines the maximum possible throughput without interference, but actual throughput at any time must consider interference. We introduce a function $K_{rr'} : [0, t_{\mathtt{max}}] \mapsto [0, 1]$ which defines the fraction of achievable goodput between agents r and r' at the input time. The actual computation for interference depends on the interference model used, and we leave the constraint general for this formulation so our analysis generalizes across all interference models.

The network flow constraints (3) expand to,

$$\sum_{r' \in \mathcal{R} \cup \mathcal{S}} v_{r'r} = \sum_{r' \in \mathcal{R} \cup \mathcal{T}} v_{rr'} \qquad \forall r \in \mathcal{R} \tag{7}$$

$$v_{rr'} \leq \frac{\int_t \mathcal{V}_{rr'}(t)\, dt}{t_{\mathtt{max}}} \qquad \forall \mathcal{D}\ r, r' \in \mathcal{A} \tag{8}$$

$$\mathcal{V}_{rr'}(t) \leq K_{rr'}(t)\, \mathcal{C}(\sigma_r(t), \sigma_{r'}(t)) \qquad \forall \mathcal{D}\ r, r' \in \mathcal{A}, 0 \leq t \leq t_{\mathtt{max}} \tag{9}$$

$$K_{rr'}(t) \leq \mathtt{InterferenceBound}(\sigma(t)) \qquad \forall \mathcal{D}\ r, r' \in \mathcal{A}, 0 \leq t \leq t_{\mathtt{max}}\ , \tag{10}$$

where (7) is a flow conservation constraint, (8) is an average flow constraint, (9) bounds the throughput based on achievable goodput, and (10) is an interference constraint. The flow conservation constraint ensures no data remains on the robot. The average flow constraint bounds the flow from one robot to another to the maximum amount of data transmitted divided by the total time—i.e., the average throughput, and the interference constraint bounds the fraction of achievable goodput based on the interference model. While (9) and (10) pose an infinite number of constraints, techniques such as [11] find and then add constraints at invalid times, converting the infinite number of constraints to a finite number.

The input flow to each sink bounds the optimal answer to RNT, and the average throughput over the entire trajectory bounds the flow into each sink (8). The amount of time it takes to move to a position is independent of the amount of data a robot holds, meaning the proportion of time a robot spends moving to any position tends to zero as $t_{\mathtt{max}}$ tends to infinity. As a consequence, the impact of such motion on the average throughput (8) trends to zero. We prove that the optimal throughput depends on the portions of the trajectory where throughput is constant rather than the throughput over time.

We divide the trajectories for any pair of agents into Constant Throughput Durations (CTDs), which are durations where the throughput is constant and non-zero for any pair of agents. Given an infinite amount of data to transmit, the throughput over the optimal trajectory is dominated by these CTDs.

Definition 2. *A Constant Throughput Duration is a portion of a trajectory* $[t_0, t_1]$ *where the instantaneous throughput between any two agents is constant and nonzero. Given a CTD,* $n = [t_0, t_1]$ *and trajectories* σ_r *and* $\sigma_{r'}$ *then*

$$K_{rr'}\left(t_0\right)\mathcal{C}\left(\sigma_r\left(t_0\right), \sigma_{r'}\left(t_0\right)\right) = K_{rr'}\left(t\right)\mathcal{C}\left(\sigma_r\left(t\right), \sigma_r\left(t\right)\right) \forall t, t_0 \leq t \leq t_1 \tag{11}$$

Assumption 1 *Robot dynamics are controllable enough and the environment is open enough such that any achievable throughput is maintainable along a static or cyclic trajectory—i.e., given a CTD* $n = [t_0, t_1]$*, there must exist a trajectory* σ_r *such that* $\sigma_r\left(t_0\right) = \sigma_r\left(t_1\right)$*.*

Assumption 1 ensures that if two robots can reach a pair of positions where the optimal answer has them transmit information, they are never forced to stop a transmission due to the dynamics of the robotics system. In practice, this typically means the robots are capable of stopping, hovering, or orbiting one another and such control is common for robot systems.

Theorem 1. *As data capacity tends to infinity, the throughput upper bound between any two agents* r, r' *tends to the weighted average of all CTDs* $\mathcal{N}$*,*

$$\lim_{d_{\mathtt{max}}\to\infty} \quad v_{rr'} \leq \sum_{t_0,t_1\in\mathcal{N}} \frac{t_1 - t_0}{t_{\mathtt{max}}} K_{rr'}\left(t_0\right)\mathcal{C}\left(\sigma_r\left(t_0\right), \sigma_{r'}\left(t_0\right)\right) \ . \tag{12}$$

Proof. As data capacity tends to infinity, constraint (6) is always satisfied thus any agent's throughput is limited by (8) and (9), which converts the throughput at every point in time to the average throughput throughout the entire trajectory. We substitute (9) into (8) bounding the average throughput based on the goodput over optimal robot trajectories, $\sigma_r, \sigma_{r'}$.

$$v_{rr'} \leq \frac{\int_t \mathcal{V}_{rr'}\left(t\right)dt}{t_{\mathtt{max}}} \leq \frac{\int_t K_{rr'}\left(t\right)\mathcal{C}\left(\sigma_r\left(t\right), \sigma_{r'}\left(t\right)\right)dt}{t_{\mathtt{max}}} \tag{13}$$

We separate both trajectories into time spent in a CTD, $\mathcal{N}$ and all other points in time, discretizing the integral.

$$\frac{\int_t K_{rr'}\left(t\right)\mathcal{C}\left(\sigma_r\left(t\right), \sigma_{r'}\left(t\right)\right)dt}{t_{\mathtt{max}}} \leq \tag{14}$$
$$\sum_{t_0,t_1\in\mathcal{N}} \frac{t_1 - t_0}{t_{\mathtt{max}}} K_{rr'}\left(t_0\right)\mathcal{C}\left(\sigma_r\left(t_0\right), \sigma_{r'}\left(t_0\right)\right) + \frac{\int_{t\notin\mathcal{N}} K_{rr'}\left(t\right)\mathcal{C}\left(\sigma_r\left(t\right), \sigma_{r'}\left(t\right)\right)dt}{t_{\mathtt{max}}}$$

By definition, if two robots are not in a CTD they must either (1) have zero throughput between them or (2) the throughput must be changing. Any achievable throughput is maintainable via Assumption 1, so changing throughput must be because the robots are moving towards but have not yet reached a CTD. Since any data that is transmitted must first be received, $t_{\mathtt{max}}$ trends towards infinity as $d_{\mathtt{max}}$ trends towards infinity. The amount of time spent moving towards a CTD is constant and thus becomes negligible as $t_{\mathtt{max}}$ tends to infinity. Therefore, as $t_{\mathtt{max}}$ tends to infinity, the contribution of cases (1) and (2) towards the throughput bound tends to zero, meaning

$$\lim_{d_{\mathtt{max}} \to \infty} v_{rr'} \leq \tag{15}$$

$$\sum_{t_0, t_1 \in \mathcal{N}} \frac{t_1 - t_0}{t_{\mathtt{max}}} K_{rr'}(t_0)\, \mathcal{C}\left(\sigma_r(t_0), \sigma_{r'}(t_0)\right) + \cancelto{0}{\frac{\int_{t \notin \mathcal{N}} K_{rr'}(t)\, \mathcal{C}\left(\sigma_r(t), \sigma_{r'}(t)\right) dt}{t_{\mathtt{max}}}}$$

□

Theorem 1 implies that given a sufficient amount of data to transmit, driving to find a location with better throughput is always optimal regardless of the path length. Examples in real-world scenarios, such as carrier pigeons carrying SD cards achieving higher throughput than broadband internet providers [31] empirically validating Theorem 1.

Theorem 1 bounds the achievable throughput; however, it does not bound other queuing theoretic metrics such as latency. The trajectory in Theorem 1 may collect an infinite amount of data from a source and then, only after all the data is collected, transmit it all to a sink, yielding an infinite latency. However, such scenarios require the robots to collect an infinite amount of data, meaning that we can ensure a finite latency by constraining a robot's storage capacity.

4.2 Relaxed Formulation

We relax our formulation for RNT to permit data transmission only during a Constant Throughput Duration, converting the SIP in Sec. 4.1 to a MIQCP. Finding an optimal solution to a SIP requires finding the set of constraints that bound the optimal solution as well as the solution itself [11], whereas solving a MIQCP does not require finding the bounding set of constraints and thus is generally easier to solve. The solution from the relaxed formulation is a sequence of trajectories that move between CTDs and a communication at the end of each trajectory. Optimal paths and transmission schedules for this relaxation will have robots either moving or transmitting but never both simultaneously. However, such trajectories are optimal if the robots have an infinite data capacity.

Theorem 1 defines the average throughput with a relationship to the proportion of time spent in a CTD—i.e., $\frac{t_1 - t_0}{t_{\mathtt{max}}}$ in (12). For each CTD, $n \in \mathcal{N}$, we define a new variable $g_n \in [0, 1]$ which is the proportion of time spent in CTD n, $k_{rr'n} \in [0, 1]$ which is the proportion of the goodput achieved between agents r and r' at CTD n, and t_n which is the start time of of CTD n.

$$\max \sum_{r \in \mathcal{R} \cup \mathcal{S}, r' \in \mathcal{T}} v_{rr'} \tag{16}$$

$$\text{s.t.} \quad \sum_{r' \in \mathcal{R} \cup \mathcal{S}} v_{r'r} = \sum_{r' \in \mathcal{R} \cup \mathcal{T}} v_{rr'} \qquad \forall r \in \mathcal{R} \tag{17}$$

$$v_{rr'} \leq \sum_{n \in \mathcal{N}} g_n k_{rr'n} \mathcal{C}\left(x_{rn}, x_{r'n}\right) \qquad \forall \mathcal{D}\ r, r' \in \mathcal{A}, n \in \mathcal{N} \tag{18}$$

$$k_{rr'n} \leq \texttt{InterferenceBound}\left(\forall r'' x_{r''n}\right) \qquad \forall \mathcal{D}\ r, r' \in \mathcal{A}, n \in \mathcal{N} \tag{19}$$

$$t_{\texttt{max}} \sum_{n \in \mathcal{N}} g_n \leq t_{\texttt{max}} - t_{\texttt{motion}} \tag{20}$$

$$\texttt{MotionBound}\left(\sigma_r, k\right) \leq t_{\texttt{motion}} \qquad \forall r \in \mathcal{R} \tag{21}$$

$$\sigma_r\left(t_n\right) = x_{rn} \qquad \forall r \in \mathcal{R}, n \in \mathcal{N} \tag{22}$$

$$\sigma_r \in \mathcal{X}_{\texttt{valid}}, \sigma_r(0) = \sigma_r\left(t_{\texttt{max}}\right) \qquad \forall r \in \mathcal{R} \tag{23}$$

$$0 \leq d_{r0} + t_{\texttt{max}} \sum_{n=1}^{n'} \sum_{r' \in \mathcal{A}} \Big(g_n \left(k_{r'rn} - k_{rr'n}\right) \mathcal{C}\left(x_{r'n}, x_{rn}\right)\Big) \leq d_{\texttt{max}} \qquad \forall r \in \mathcal{R}, n \in \mathcal{N} \tag{24}$$

Where (17), (19), (23), and (24) are the same constraints as (7), (10), (5), and (6); (18) is the throughput limit derived from Theorem 1; and (20) constrains the proportion of time spent in each CTD relative to the total and motion time. Constraint (21) constrains motion time based on the time it takes to traverse a trajectory, while also allowing robots to move without adding to $t_{\texttt{motion}}$ if they move in a CTD that they do not communicate during.

Theorem 2. *Our relaxed formulation* (16) *converges to the optimal network throughput for RNT as data capacity tends to infinity.*

Proof. From Theorem 1, the upper bound on throughput between any two agents tends solely to the throughput at any CTD. Our relaxed formulation only considers data transfers at CTD. Therefore, an optimal solution to our relaxed formulation is bounded by Theorem 1's upper bound as data capacity tends to infinity and is an optimal solution to RNT. □

5 Bottleneck Path Upper Bound

We construct a Robot Network Graph (see Fig. 2) representing an upper bound for RNT and use it in conjunction with the relaxed formulation (Sec. 4.2) to find a bounded optimal network throughput to RNT. Our Robot Network Graph represents throughput due to robot motion and communication, allowing us to reason about both simultaneously. By jointly representing throughput from motion and wireless communication, we derive tight bounds offering insight into whether wireless throughput or robot motion is the network bottleneck.

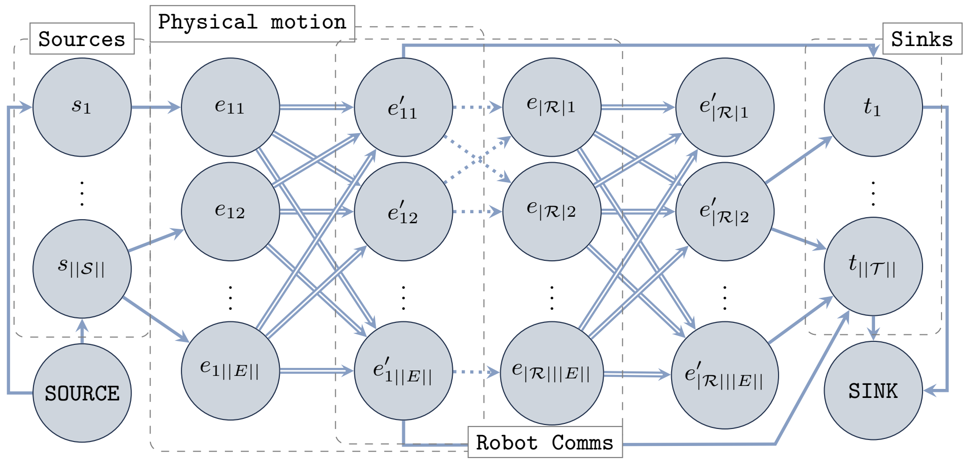


Fig. 2: The Robot Network Graph representing data flow through motion (shown as double lines) or wireless communication (shown as single lines) used to find the network throughput upper bound. The physical motion subgraphs are connected by edges representing robot-to-robot communication. We find an network throughput upper bound by evaluating the bottleneck paths Sec. 5.3 from `SOURCE` to `SINK`, which has polynomial runtime.

Definition 3. *A* <u>*bottleneck path*</u> *is a path from a start vertex to a destination vertex through a graph that maximizes the minimum edge weight along the path.*

Under Definition 3, the minimum edge weight on the bottleneck path represents the network's "bottleneck" that limits overall throughput [3].

Definition 4. *A* <u>*Robot Network Graph*</u> *(see Fig. 2) is a weighted directed acyclic graph representing data flow through the robots' environment. We are given a network with a finite set of sources $\mathcal{S}$, a finite set of sinks $\mathcal{T}$, and an environment with a finite set of regions $\mathcal{E}$. The graph has vertices,*

- *for each source ($s \in \mathcal{S}$);*
- *for each sink ($t \in \mathcal{T}$);*
- *for receiving at each region (e_{ij}, $\forall i \in \mathcal{R}$, $\forall j \in \mathcal{E}$);*
- *for transmitting at each region (e'_{ij}, $\forall i \in \mathcal{R}$, $\forall j \in \mathcal{E}$);*
- *for a virtual* `SOURCE` *and* `SINK`*.*

The graph has edges,

- *from each source to initial receiving vertices $s \to e_{1i}$, $\forall i \in \mathcal{E}$ $\forall s \in \mathcal{S}$;*
- *from each transmission vertex to each sink $e'_{ij} \to t$ $\forall i \in \mathcal{R}, \forall j \in \mathcal{E}$ $\forall t \in \mathcal{T}$;*
- *from each reception vertex to corresponding transmission vertex, $e_{ij} \to e'_{ij'}$, $\forall i \in \mathcal{R}$ $\forall j, j' \in \mathcal{E}$, representing the throughput due to physical motion;*
- *from each transmission vertex to sequential reception vertex, $e'_{ij} \to e_{i+1j'}$, $\forall i \in \mathcal{R}$ $\forall j, j' \in \mathcal{E}$, representing robot-to-robot communication;*
- *from the virtual source to each source* `SOURCE` *$\to s$ $\forall s \in \mathcal{S}$ with infinite weight;*

– *from all sinks to the virtual sink* $t \to \texttt{SINK}\ \forall t \in \mathcal{T}$ *with infinite weight.*

We consider paths starting at the virtual `SOURCE` and ending at the virtual `SINK`, finding paths that go through any source-sink pair. Furthermore, the path flows through one or more vertices e_{ij}. Traversing e_{ij} means that data has reached the i^{th} robot in the chain located in region j.

The maximum flow through Robot Network Graph is an upper bound on network throughput for RNT when the edge weights are the throughput between regions; however classic max-flow does not constrain the number of visited vertices, meaning the max-flow may require more robots than are on the team. We introduce a Constrained Max-Flow problem where finding the throughput the Robot Network Graph would be a tighter network throughput upper bound than an unconstrained max-flow.

Definition 5. *Given a graph G, vertex set, V, and capacity c, a Constrained Maximum Flow is the maximum flow through graph G while visiting at most c vertices in V.*

Proposition 1. *Solving a Constrained Maximum Flow problem over Robot Network Graph with vertex set* $\{e_{ij}, e'_{ij} \forall i \in \mathcal{R}\ \forall j \in \mathcal{E}\}$ *and capacity* $2\,|\mathcal{R}|$ *is an upper bound on network throughput.*

We find a bound for the flow in Prop. 1 by greedily finding bottleneck paths that maximize the rate flow increases relative to the number of vertices visited—i.e., the rate of throughput increases relative to the number of robots. We show that the bound created by maximizing the rate at which flow increases must be a bound for the problem in Prop. 1 and thus the network throughput of RNT.

Our Robot Network Graph represents upper bounds on network throughput rather than the exact optimum due to three simplifying assumptions to construct the graph. Each assumption strictly increases the possible network throughput, meaning analyzing the graph produces upper bounds on network throughput. We show that the upper bound produced by the flow graph is tight, even for complex problems where the solution violates the assumptions in our experiments Sec. 6.

Assumption 2 *Robots move and communicate between discrete convex regions, rather than continuous positions in the environment.*

Any environment can be decomposed into discrete convex regions via Delaunay triangulation [18], so this assumption does not limit environments where we can construct the Robot Network Graph. Moreover, further discretization increases the bound's accuracy.

Assumption 3 *The only interference is self-interference—i.e., a robot transmitting data interfering with its ability to receive data.*

Finding network flow for static nodes is NP-hard considering interference [8] and finding flow through a Robot Network Graph is an almost linear time problem [4], meaning network flow cannot be used to account for interference with a polynomial number of nodes unless P=NP.

Assumption 4 *There is always a robot to receive data.*

Our Robot Network Graph represents data flow between regions of space, assuming that there is always a robot available to communicate. We relax this assumption when finding the network throughput upper bound, reasoning about the number of robots required to any flow.

We modify the edge weights of the Robot Network Graph to represent the upper bound on the throughput increase per robot, which we use to bound the Constrained Maximum Flow. We describe the communication throughput limit considering self-interference in Sec. 5.1 and robot motion throughput in Sec. 5.2. Combining the insights from both, we prove that our Robot Network Graph is an upper bound in Sec. 5.3 and show the bound's tightness in Sec. 6.

5.1 Communication Throughput

We add communication edges (shown as single lines in Fig. 2) with capacities based on the maximum goodput between regions, ensuring that the flow along any edge is an upper bound on the per-robot throughput. Furthermore, the Robot Network Graph has vertex capacities based on the maximum goodput for any robot since robots experience self-interference per Assumption 3.

Communication Edge Capacity Communication edge capacities represent the maximum possible goodput increase per-robot between two regions (per Assumption 2). Since each communication edge has at least one vertex representing a region, we ensure the edge capacity is a valid upper bound by finding the maximum per-robot goodput between anywhere in the region.

All source vertices connect to the first physical motion subgraph, only requiring one robot. Therefore, given a source vertex $s \in \mathcal{S}$ and region $e \in \mathcal{E}$ the maximum per-robot goodput is the maximum goodput, $\mathcal{C}_{se}$. Sink vertices, conversely, connect to each physical motion subgraph and each successive physical motion subgraph represents another required robot. Therefore, given a sink vertex $s \in \mathcal{T}$, region $e \in \mathcal{E}$, and a motion subgraph index $i \in [1, |\mathcal{R}|]$, the maximum per-robot goodput is the maximum goodput, $\mathcal{C}_{se}$, divided by the number of required robots, $\frac{\mathcal{C}_{se}}{i}$.

Communication edges representing robot-to-robot communication connect the sequential physical motion subgraphs of the Robot Network Graph, meaning the pair of robots anywhere in either region. From Assumption 4, we assume there are always robots to receive information, meaning given regions $e_0, e_1 \in \mathcal{E}$ with maximum goodput $\mathcal{C}_{e_0 e_1}$ between them the edge capacity between motion subgraphs i and $i+1$ is, $\frac{\mathcal{C}_{e_0 e_1}}{i+1}$.

Vertex Capacity We add vertex capacities for all vertices, representing the maximum possible throughput increase any motion or transmissions could have. We maximize the throughput through the source and sink nodes when they constantly transmitting or receiving, meaning that every source and sink node has vertex capacity $\mathcal{C}_{\texttt{max}}$. Conversely, every robot must transmit all information it receives and must contend with self-interference (per Assumption 3), which means every vertex in physical motion subgraph i has the capacity, $\frac{\mathcal{C}_{\texttt{max}}}{2i}$.

5.2 Robot Motion Throughput

We add motion edges (shown as thick lines in Fig. 2) between region vertices representing the throughput increase due to a robot moving between regions. Throughput is the rate at which data moves between positions, so motion throughput is bounded by the data capacity of a robot and the time it takes to move. Given any two regions $e_0, e_1 \in \mathcal{E}$ and the time it takes to move between them $t_{e_0 e_1}$, the per-robot throughput of any edge in physical motion subgraph i is the data capacity divided by the time to perform a cyclic path $\frac{d_{\texttt{max}}}{i\left(t_{e_0 e_1} + t_{e_1 e_0}\right)}$.

5.3 Robot Network Graph Analysis

We find bottleneck paths through the Robot Network Graph (Fig. 2), constructing a network throughput upper bound based off of the throughput increase per-robot and the number of robots in the team. Our bound requires us to find at most $|\mathcal{R}|$ paths, which can be done in polynomial time [3]. Furthermore, since our Robot Network Graph describes motion between regions of space, the upper bound also corresponds to a trajectory, which we use to create a lower bound.

Algorithm 1: Bottleneck Path Upper Bound

Input: Σ // Definition 1
Output: σ, b // Heuristic start, and upper bound
1 $G \leftarrow$ `constructGraph` (Σ) // Fig. 2
2 $b, R \leftarrow 0, |\mathcal{R}|$ // Bound and available robots
3 **while** $R > 0$ **do**
  // Bottleneck path through Fig. 2 and per-robot throughput
4 $p, b' \leftarrow$ `bottleneckPath` (G) // [3]
  // Available and requested robots
5 $R_{\texttt{need}}, R_{\texttt{want}} \leftarrow$ `requiredRobots` $(p), \max_{s \in p} \frac{\texttt{vertexCapacity}(p)}{b'}$
6 $b'' \leftarrow \max\left(1, \texttt{intDivide}\left(R_{\texttt{want}}, R_{\texttt{need}}\right)\right) R_{\texttt{need}}$ // Find network throughput
  // Reduce vertex capacity of source and sink
7 **for** $s \in$ `staticNodes` (p) **do** `vertexCap` $(G, s) \leftarrow$ `vertexCap` $(G, s) - b''$
  // Remove the used robots and increase the bound
8 $R, b \leftarrow R - \left(\frac{b''}{b'}\right), b + b''$
  // Add heuristic start if there are available robots
9 **if** $R \geq 0$ **then** $\sigma \leftarrow \sigma \cup$ `graphToContinuousPath` (p)

Upper Bound We construct a network throughput upper bound from the Robot Network Graph (Fig. 2), finding bottleneck paths [3] representing trajectories for portions of the team (see Alg. 1). We iteratively find the best bottleneck path (line 4) and associated robots (line 5), assigning each robot a trajectory until there are no more robots remaining. A bottleneck path through Robot Network Graph represents the maximum rate at which throughput can increase with respect to number of robots, and we show the sum of rate increases multiplied by robots is an upper bound on Prop. 1 and thus the network throughput.

Theorem 3. *Given a Constrained Max Flow Problem with vertex capacity V and an ordered sequence $\mathcal{B} = \left[\left(b'^{\langle 0\rangle}, v_{\mathtt{need}}^{\langle 0\rangle}\right), \ldots, \left(b'^{\langle|\mathcal{B}|\rangle}, v_{\mathtt{need}}^{\langle|\mathcal{B}|\rangle}\right)\right]$ where $b'^{\langle i\rangle}$ is the rate flow increases per vertex on a path, $v_{\mathtt{need}}^{\langle i\rangle}$ is the number of vertices on the same path, and $b'^{\langle i\rangle} \geq b'^{\langle i+1\rangle}$ $\forall i \in |\mathcal{B}|$ then the Constrained Max Flow is bounded by*

$$b \triangleq \sum_{i=0}^{|\mathcal{B}|} \begin{cases} b'^{\langle i\rangle} v_{\mathtt{need}}^{\langle i\rangle}, \text{ if } \sum_{j=0}^{i-1} v_{\mathtt{need}}^{\langle j\rangle} < V \\ 0, \text{ otherwise} \end{cases} . \tag{25}$$

Proof. b is the flow from greedily summing the flow from the paths with the highest flow increase per-vertex until the number of vertices visited is greater than the capacity. Since rates in $\mathcal{B}$ are monotonically decreasing, b must be greater than the flow from any constrained set.

$$b \geq \sum_{b', v_{\mathtt{need}} \in \mathcal{B}'} b' v_{\mathtt{need}} \ \forall \mathcal{B}' \in 2^{\mathcal{B}}, \sum_{b', v_{\mathtt{need}} \in \mathcal{B}'} v_{\mathtt{need}} \leq V \tag{26}$$

Since b is bigger than any valid Constrained Max Flow it must be an upper bound on flow. □

Theorem 4. *Alg. 1 is an upper bound network throughput for RNT.*

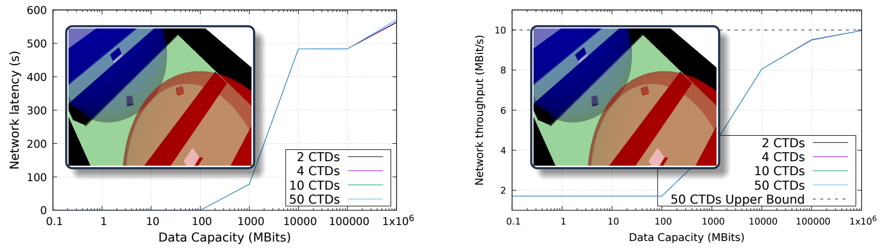
Proof. From Prop. 1, a Constrained Max Flow problem is an upper bound on network throughput, and Theorem 3 shows that Constrained Max Flow problems are bounded by the greedily summing the flow increases. Alg. 1 finds the bottleneck paths that maximize the rate of throughput increase relative to the number of robots, calculating b in Theorem 3, meaning it is an upper bound for Prop. 1. Therefore Alg. 1 is a network throughput upper bound. □

Runtime Our upper bound (see Alg. 1) runs in polynomial time. Bottleneck path algorithms run in linear time with the number of edges [3], and the while-loop (line 3) can at most run $|\mathcal{R}|$ times, yielding $\mathcal{O}\left(|\mathcal{R}|\left(|\mathcal{S}|\,|\mathcal{E}| + |\mathcal{R}|\,|\mathcal{E}|^2 + |\mathcal{R}|\,|\mathcal{E}|\,|\mathcal{T}|\right)\right)$.

Lower Bound Our upper bound produces robot trajectories based on the bottleneck path found (line 9). Therefore, every upper bound has an associated trajectory beginning and ending in each region described by the path through the Robot Network Graph, which we use as extra constraints in our optimization formulation Sec. 4.2 producing a lower bound on network throughput in RNT.

6 Experiments

We perform simulated RNT experiments showing how the throughput of motion changes with data capacity. Furthermore, we analyze the optimality of our relaxed formulation (Sec. 4.2) as we increase the number of CTDs, as well as the trade-off between network throughput and network latency. We show the tightness of our upper-bound (Sec. 5) as team size increases.



(a) Network Latency (b) Network Throughput

Fig. 3: The environment for the two robot tests in Sec. 6.1. The source is shown in blue at the top, the sink at the bottom in red, and communication linearly degraded with distance. Two robots moved data from the source to the sink, and from Theorem 1, we would expect the optimal network throughput to involve taking the longer path around the barriers as data capacity trends to infinity, which is what we observed. Similarly, as data capacity increases the optimal network throughput trends towards a store-and-forward approach, increasing latency and implying that by controlling data capacity we can control latency.

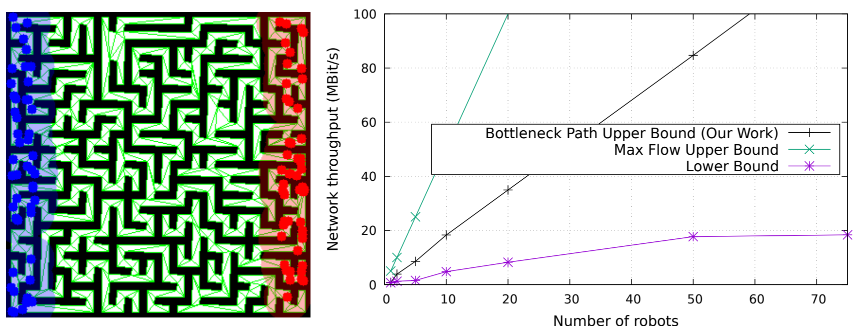


Fig. 4: The maze environment and results for our scalability analysis in Sec. 6.2, with environment partitioning shown in green. The fifty source and sink nodes are shown on the left and right respectively with their communication ranges. Our upper bound (Bottleneck Path Upper Bound) closer to the lower bound compared to a typical max flow upper bound.

We performed optimization with Gurobi [9] and used Delaunay Triangulation to decompose the environment [18]. We enforced interference the same as [8], ensuring that robots can only transmit if noise from other transmissions is below a certain threshold. For all experiments, the maximum goodput was $10\,\mathrm{Mbit\,s^{-1}}$.

6.1 Performance Bounds of Sec. 4.2

We evaluate the network throughput of our relaxed formulation (Sec. 4.2) as data capacity increases with a two-robot experiment in Fig. 3. For this experiment, communication linearly degraded with distance, meaning traveling around either barrier would result in better instantaneous throughput but with more motion time. We report network throughput and latency after optimizing network throughput using (16) for one hour. We increased each robot's data capacity and number of CTDs, showing network throughput approaches the bound in Theorem 1 as capacity increases regardless of the number of CTDs used.

Our results in Fig. 3 confirm our theoretical claims that as data capacity approaches infinity, the network throughput approaches only the throughput at a CTD. Adding more CTDs increases the difficulty of optimization and after an hour the methods with > 2 CTDs still had an optimality gap, meaning performance could still potentially be improved. However, the lower bound for every method approached the network throughput upper bound of $10\ \frac{\text{Mbit}}{\text{s}}$ as data capacity increased. Convergence to the upper bound follows Theorem 1, which bounds network throughput by only the throughput during CTDs.

6.2 Scalability Analysis

We analyze the tightness of our Bottleneck Path upper bound on network throughput by comparing it to the maximum flow across the Robot Network Graph and the heuristic minimum network throughput as the robot team size increases. We used a constant communication model for this experiment, 50 sources, 50 sinks, 584 regions, and 2 CTDs when computing the lower bound. The sources can interfere with at least one other, invalidating Assumption 3 of our upper bound. Despite this, our Bottleneck Upper Bound was at least 3x tighter than the Max Flow upper bound when team size grew beyond 20 robots (see Fig. 4), which scaled rapidly due to max flow finding potential trajectories that used more robots than were available.

While our Bottleneck Path upper bound produced a tighter bound max flow, the optimality gap between the heuristic lower bound grew as team size increased, implying either our lower or upper bound got worse as team size increased. The lower bound network throughput staying constant after 50 robots happens because of network interference at the cluttered sources and sinks, implying that with lower numbers of robots, the network interference was lower, meaning assumptions Assumption 3 and Assumption 4 were met. Future work will focus on tightening both the lower and upper bound, decreasing the optimality gap as team size increases.

7 Conclusion

We analyzed the Robot Network Throughput problem where a team of robots maximizes network throughput. We found bounds on throughput due to robot motion. Using our insights, we formulated a Mixed Integer Quadratically Constrained Program (MIQCP) that finds the optimal answer to RNT as each robot's data capacity trends towards infinity and an upper bound based on bottleneck paths considering motion and communication. By combining the MIQCP lower bound and bottleneck paths upper bound, we created a bounded-optimal algorithm for finding the maximum network throughput. We showed upper bound on network throughput stayed tighter than a typical max-flow upper bound in complex environments with many robots, sources, and sinks. Future work will focus on different ways to generate bounds on network throughput.

References

1. Altman, E., Neglia, G., De Pellegrini, F., Miorandi, D.: Decentralized stochastic control of delay tolerant networks. In: IEEE INFOCOM 2009. pp. 1134–1142. IEEE (2009)
2. Bhadauria, D., Tekdas, O., Isler, V.: Robotic data mules for collecting data over sparse sensor fields. Journal of Field Robotics **28**(3), 388–404 (2011)
3. Camerini, P.M.: The min-max spanning tree problem and some extensions. Information Processing Letters **7**(1), 10–14 (1978)
4. Chen, L., Kyng, R., Liu, Y.P., Peng, R., Gutenberg, M.P., Sachdeva, S.: Maximum flow and minimum-cost flow in almost-linear time. In: 2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS). pp. 612–623. IEEE (2022)
5. Diller, J., Hall, P., Han, Q.: Holistic path planning for multi-drone data collection. In: 2023 19th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT). pp. 222–226. IEEE (2023)
6. Edmonds, J., Karp, R.M.: Theoretical improvements in algorithmic efficiency for network flow problems. Journal of the ACM (JACM) **19**(2), 248–264 (1972)
7. Ford, L.R., Fulkerson, D.R.: Maximal flow through a network. Canadian journal of Mathematics **8**, 399–404 (1956)
8. Gupta, P., Kumar, P.R.: The capacity of wireless networks. IEEE Transactions on information theory **46**(2), 388–404 (2000)
9. Gurobi Optimization, LLC: Gurobi Optimizer Reference Manual (2023), `https://www.gurobi.com`
10. Harris, T., Ross, F.: Fundamentals of a method for evaluating rail net capacities. Tech. rep., Rand Corporation (1955)
11. Hauser, K.: Semi-infinite programming for trajectory optimization with non-convex obstacles. The International Journal of Robotics Research **40**(10-11), 1106–1122 (2021)
12. Huang, H., Savkin, A.V.: Viable path planning for data collection robots in a sensing field with obstacles. Computer Communications **111**, 84–96 (2017)
13. Huang, M., Chen, S., Zhu, Y., Wang, Y.: Topology control for time-evolving and predictable delay-tolerant networks. IEEE Transactions on Computers **62**(11), 2308–2321 (2012)
14. Jain, K., Padhye, J., Padmanabhan, V.N., Qiu, L.: Impact of interference on multihop wireless network performance. In: Proceedings of the 9th annual international conference on Mobile computing and networking. pp. 66–80 (2003)
15. Jain, S., Fall, K., Patra, R.: Routing in a delay tolerant network. In: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications. pp. 145–158 (2004)
16. Jones, E.P., Li, L., Ward, P.A.: Practical routing in delay-tolerant networks. In: Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking. pp. 237–243 (2005)
17. Kantaros, Y., Guo, M., Zavlanos, M.M.: Temporal logic task planning and intermittent connectivity control of mobile robot networks. IEEE Transactions on Automatic Control **64**(10), 4105–4120 (2019)
18. Lo, S.: Delaunay triangulation of non-convex planar domains. International Journal for Numerical Methods in Engineering **28**(11), 2695–2707 (1989)
19. Marcucci, T., Umenberger, J., Parrilo, P., Tedrake, R.: Shortest paths in graphs of convex sets. SIAM Journal on Optimization **34**(1), 507–532 (2024)

20. Newman, D.: Benchmarking Terminology for Firewall Performance. RFC 2647 (Aug 1999). https://doi.org/10.17487/RFC2647, https://www.rfc-editor.org/info/rfc2647
21. Schack, M.A., Rogers, J.G., Han, Q., Dantam, N.T.: Optimization-based robot team exploration considering attrition and communication constraints. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE/RSJ (2021)
22. Schack, M.A., Rogers, J.G., Han, Q., Dantam, N.T.: Robot team data collection with anywhere communication. In: 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE/RSJ (2023)
23. Scherer, J., Rinner, B.: Multi-UAV surveillance with minimum information idleness and latency constraints. IEEE Robotics and Automation Letters **5**(3), 4812–4819 (2020)
24. Scherer, J., Schoellig, A.P., Rinner, B.: Min-max vertex cycle covers with connectivity constraints for multi-robot patrolling. IEEE Robotics and Automation Letters **7**(4), 10152–10159 (2022)
25. Schulman, J., Duan, Y., Ho, J., Lee, A., Awwal, I., Bradlow, H., Pan, J., Patil, S., Goldberg, K., Abbeel, P.: Motion planning with sequential convex optimization and convex collision checking. The International Journal of Robotics Research **33**(9), 1251–1270 (2014)
26. Scott, K., Burleigh, S.: Bundle protocol specification. Tech. rep. (2007)
27. Shannon, C.E.: Communication in the presence of noise. Proceedings of the IRE **37**(1), 10–21 (1949)
28. Tanenbaum, A.S.: Computer networks. Pearson Education India (2003)
29. Yu, X., Hsieh, M.A.: Synthesis of a time-varying communication network by robot teams with information propagation guarantees. IEEE Robotics and Automation Letters **5**(2), 1413–1420 (2020)
30. Zhao, W., Ammar, M., Zegura, E.: Controlling the mobility of multiple data transport ferries in a delay-tolerant network. In: Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies. vol. 2, pp. 1407–1418. IEEE (2005)
31. Zyga, L.: Carrier pigeon faster than broadband internet, https://phys.org/news/2009-09-carrier-pigeon-faster-broadband-internet.html