Throughput-Optimal Robotic Message Ferrying for Wireless Networks using Backpressure Control

Andrea Gasparri
Department of Engineering
Roma Tre University
Rome, Italy
gasparri@dia.uniroma3.it

Bhaskar Krishnamachari
Dept. of Electrical Engineering
University of Southern California
Los Angeles, CA
bkrishna@usc.edu

***Abstract*—We consider the problem of controlling the motion of a set of robots to ferry messages between a given set of statically-placed nodes. The design and analysis of an arrival-rate unaware throughput-optimal policy for this problem is challenging because of the coupling between position and link rate. We propose a fine-grained backpressure message ferrying algorithm (FBMF) for joint motion and transmission control of robots. Unlike traditional backpressure settings, because the controlled motion of the relay nodes changes the channel rates, it turns out that the conventional approach to prove throughput optimality does not work in this problem setting. We prove for the simplest setting (single-flow, single-robot, constant arrival) that this policy indeed achieves throughput optimality. The analysis reveals that under feasible traffic, even when queues are highly over-loaded, the change in the total queue size can be positive over a time step, nevertheless the system exhibits a limit-cycle behavior and stability holds because the change in the total queue size is negative over the cycle for sufficiently large queues. We pose the design and analysis of a throughput optimal policy for the general case as a challenging open problem for network theory.**

Keywords*—*Mobile Sensor Networks, Backpressure Algorithm, Throughput-Optimality.

I. INTRODUCTION

Recently, there have been a number of developments in the theory and practical realization of distributed multi-agent robotics. It is becoming increasingly affordable to deploy teams of robots in an operational environment, in order to perform desired sensing, communication and other tasks in a coordinated fashion. These developments open up the possibility of using robotic relays to enhance the performance of wireless networks.

Over the past two decades, wireless networking research has largely focused on *unpredictable and uncontrollable* mobility (typically, due to motion of humans carrying the mobile devices network), and to a lesser extent on *predictable but uncontrollable* mobility (applicable whenever nodes follow a somewhat predictable schedule such as buses and trains). One perspective on the novelty and benefit of introducing robotic relays in the context of wireless networks is that they provide a new design dimension, that of *controllable* (and *ipso facto*, *predictable*) mobility, which can be used to enhance data routing.

In this work, we consider a simple setting where controlled mobility would be clearly beneficial — to help ferry data between static wireless nodes that are distant from each other. Such a scenario could arise, for instance, in the context of communication in a tactical wireless network (such as for disaster relief or remote exploration) or sparsely-deployed sensor networks, where there are a limited number of nodes distributed over a large field. Applications that may involve the scenario considered in this paper include sensing and communication in remote environments for planetary exploration, wildlife monitoring, ocean science, and security/surveillance.

More concretely, in the scenario we consider there are unicast flows consisting of static source-sink pairs, and robots that may be used to ferry data between them. In a recent work [23], we show that the throughput capacity region (set of all feasible arrival rate vectors for which all queues can be stabilized) for such a system can be characterized in closed form and that any feasible arrival rate can be served by scheduling a suitably chosen convex combination of "basis" allocations consisting of one or two robots ferrying between the source and sink for each flow. However, if the arrival rate vector is not *a priori* known, it turns out to be a harder problem to schedule robots to ensure stability. In [23], we gave a scheduling algorithm for coarse-grained backpressure message ferrying (CBMF) in which robots are allocated to move towards and communicate with sources and sinks based on a queue-differential max-weight matching once every epoch, where an epoch consists of multiple slots.

While the algorithm is conceptually similar to the classic Backpressure solution first identified by Tassiulas and Ephremides [1], there is an important wrinkle in the message ferrying problem — the control at each epoch not only schedules links but also changes the robots' positions and hence their communication rates. Traditional proofs of throughput-optimal scheduling using backpressure assume that any stochasticity in the channel states / networks topology is independent of the scheduling decisions made at each step.

It turns out in fact that the CBMF algorithm from [23] is not throughput optimal for any fixed epoch size; we were able to show only that for a fixed epoch size, CBMF can stably serve arrival rates within a well-defined subset of the capacity region. As the epoch size is increased, the stability region for increases, asymptotically reaching the full capacity region. However, this comes at the cost of significantly higher delay for lower-rate traffic. This motivates us to identify a more effective solution for scheduling message ferrying robots.

In this paper, we identify such a solution, which we refer to as fine-grained backpressure based message ferrying (FBMF). In this algorithm, which is also conceptually similar to traditional backpressure, the matching of robots to source/sink is done at each time slot based on the current queue conditions and the current link rates. What we find empirically is that the fine-grained backpressure policy results in the robots eventually settling down to a limit cycle such that for sufficiently loaded queues the drift in total queue size is negative *over the whole period* of the limit cycle. Although, because actions affect channel rates in this problem, we find it difficult to analyze the general case, we present a detailed analysis for the simplest setting of a single robot serving a single flow, under constant-rate arrivals.

To summarize, the key contributions of this work are:

- A further investigation of the novel problem of joint motion and transmission control for robotic message ferrying, significantly going beyond our prior work in [23].
- Proposal of the novel fine-grained backpressure-based message ferrying (FBMF) algorithm for arrival-unaware scheduling of robot motion and transmission.
- Mathematical proof for the single-flow, single-robot setting that the proposed algorithm is throughput optimal. The proof is novel, as it shows the existence of periodic motion when queues are sufficiently loaded, such that the robot automatically moves from the source to the sink and back, resting at each end just long enough to ensure stability.
- Numerical simulations of FBMF indicating its throughput optimality and its superiority to the Coarse grained CBMF algorithm proposed in [23].

The rest of the paper is organized as follows. In Section II relate work is reviewed. In Section III the problem formulation is provided. In Section IV the prior work we developed for this problem setting is briefly reviewed. In Section V the proposed fine-grained backpressure-based message ferrying algorithm is described and the proof of throughput optimality for the simplest setting of a single-flow, single-robot is given. In Section VII simulations results to corroborate the theoretical finding are described, and finally in Section VIII conclusions are drawn and future work is discussed.

II. Related Work

The original paper on queue backpressure-based max weight scheduling by Tassiulas and Ephremides [1] showed that it stabilize any arrival rate vector in the capacity region of a general network. Later, a number of works showed how the method of Tassiulas and Ephremides can also be incorporated with utility maximization in networks [2], [4], [3].

Many recent works on backpressure scheduling have focused on enhancing delay performance. Backpressure scheduling and routing has been improved using shadow queues to handle multicast sessions [6] and improve scalability by reducing the number of queues to be maintained [6], [7]. Recent work has also shown how to use a cumulative time packet age queue formulation to reduce delay substantially [9]. Backpressure using LIFO service has been shown to offer better delay performance [10]. The framework has been extended to handle finite buffer sizes [8]. Other researchers have focused on how to make backpressure scheduling more distributed so that it can be implemented more easily [13]. More recently, there have been several reductions of backpressure theory to practice, in the form of practically implemented and experimentally evaluated distributed protocols [14], [15], [16].

A common theme to nearly all the theoretical papers on backpressure theory to date, going back to the original paper by Tassiulas and Ephremides is that the policy is typically derived by the minimization of a bound on the drift of the Lyapunov function [5]. However, all the prior approaches assume that the channel states evolve stochastically independently of the scheduling actions. In this work, this is no longer the case, as the motion control decisions for the robot (which are part of the scheduling actions) directly affect the rates to the source and sink nodes for the robot, and hence affect the channel state.

With respect to mobility, the original backpressure algorithm also applies to networks with stochastic, uncontrolled mobility. But researchers have explored how to improve its delay performance. In [11], a backpressure plus source routing approach is used for reducing the delay for intermittently connected mobile networks. And a novel adaptive redundancy technique is presented in [12] to improve the low-rate delay performance of backpressure in intermittently connected mobile networks. However, these prior works again do not address the improvement of delay in settings with controlled mobility, the focus of this work.

The integration of robotics and wireless networking is an emerging domain. Researchers have previously investigated deploying mobile nodes to provide sensor coverage in wireless sensor networks [17], [18]. In [21], [19], [22], the authors present techiques to control a teams of wireless relays to ensure connectivity and optimize network goals such as minimizing routing costs. Going beyond connectivity, recently, research has also addressed how to control a team of robots to maintain certain desired end-to-end rates while moving robots to do other tasks, referred to as the problem of maintaining network integrity [20]. In [20], the authors interleave a primal-dual-based rate allocation algorithm with potential-based robotic motion control.

To our knowledge, the only prior work on queue-aware joint robotic control and transmission scheduling is our own recent workshop submission [23]. In [23] we consider the same problem of scheduling message ferrying robots between a given set of sources and sinks. The focus of that work is to introduce, analyze, and evaluate a coarse-grained backpressure-based message ferrying scheme in which time slots are grouped into epochs of a fixed size T, and for the entire epoch duration robots are committed either to moving towards and filling-up from a source, or moving towards and draining to a given sink (by a schedule that aims to maximize queue-diffential weight). We showed that for any fixed T, CBMF is not throughput optimal, and can only stabilize arrival rates within a strict subset of the capacity region. As T is increased, the servicable region grows asymptotically towards the capacity region, however, the delay even for low rates gets arbitrarily bad (scaling linearly with T). This has motivated us to propose and investigate the novel fine-grained backpressure (FBMF) algorithm in this paper.

III. Problem Formulation and Modeling

Time is divided into unit increments. We also discretize and represent the operational area by a connected graph $G = (V, E)$ where nodes represent possible locations and edges between them represent the possibility of moving between these positions in a unit time step. There are K unicast flow pairs of static source and destination nodes. The sources are labelled as $src_1 \ldots src_K$, and the sink corresponding to source i is labelled as $sink_i$. The sources and sinks are each located on different vertices of the graph. There is a rate λ_i at which packets arrive at source i, and the arrival rate vector is $\lambda = (\lambda_1, \ldots \lambda_K)$.

There is a set of $N \leq 2K$ mobile robotic nodes that act as message ferries. These may move to any vertex on the graph (including those labelled with source or sink positions or those where other robots are present), but are only allowed to move from one vertex to the neighboring node on the graph in a unit time. They are able to communicate using radios with each of the source and sink nodes in the network at different throughput rates [1] depending on their location. Let $R_{ij}^{src}(t), R_{ij}^{sink}(t)$ represent the position-dependent rate obtained between a robot j and source i, and between robot j and sink i respectively, at time t. We assume that the rate is strictly positive, monotonically decreasing with distance, with a maximum value of R_{max} obtained when the robot is at the same vertex as the corresponding source/sink. We assume simple half-duplex radios at each node so that each node can either send or receive at any time, and communicate with only one node at a time. Each robot maintains K queues, one for each flow, labelled $Q_j^i(t)$. Each robot is capable of communicating to any source or sink on an orthogonal channel (e.g. frequency channel or spread spectrum code) from the other robots so that there is no interference between them; at any time at most one robot can be in communication with any source or sink.

We now review the concept of throughput optimality, a fundamental property of backpressure based scheduling routing policies. The reader is referred to [5] for a comprehensive treatment of the topic.

***Definition** 1 (Throughput Optimality):* A routing policy is said to be throughput optimal if it stabilizes the network for all arrival rate matrices that belong to the interior of the stability region.

We reiterate that our goal in this paper is to introduce the fine-grained backpressure message ferrying algorithm (FBMF) for joint motion and transmission control of robots described in Section V and prove that it indeed *achieves* throughput optimality for the simplest setting (single-flow, single-robot, constant arrival). The analysis reveals that under feasible traffic, even when queues are highly over-loaded, the change in the total queue size can be positive over a time step, nevertheless the system exhibits a limit-cycle behavior and stability holds because the change in the total queue size is negative over the cycle for sufficiently large queues.

[1] We use the terms throughput and rate interchangeably in this paper. We are agnostic to how these rates are made to vary at different location; e.g., via adaptive modulation schemes, or variable rate codes or simply due to differing link loss rates in case of fixed-transmission rate radios.

IV. Summary of Prior Work

In [23] we have shown the following result, which we summarize briefly in this section only for the sake of completeness (all content after this section is unique to this paper):

***Theorem** 1:* Define $\mathbf{\Lambda}$ as the following set of arrival rates:

$$\mathbf{\Lambda} = \left\{ \lambda | 0 \leq \lambda_i < R_{max}, \ \forall\ i, \ \sum_{i=1}^{K} \lambda_i < \frac{R_{max}\, N}{2} \right\}$$

Λ is the capacity region of the network for the robotic message ferrying problem. In other words, any arrival rate in this region can be stably served.

Specifically, we showed that this arrival rate region $\mathbf{\Lambda}$ can be served by a convex combination of configurations in which robots are allocated to serve distinct flows.

Let $\tilde{\Gamma}$ be a finite set of vectors defined as:

$$\tilde{\Gamma} = \left\{ \gamma | \gamma_i = \frac{a_i\, R_{\max}}{2}, \quad \forall\ i, a_i \in \{0, 1, 2\}, \sum_{i=1}^{K} a_i \leq N \right\}.$$

For each element of this set $\tilde{\Gamma}$ the corresponding integer vector $\mathbf{a}$ corresponds to a "basis" ideal allocation of robots (ignoring transmit time) to distinct sources and sinks that can service each flow at rate γ_i. Specifically, a_i refers to the number of robots allocated to service flow i. Let us refer to the convex hull of $\tilde{\Gamma}$ as $\mathcal{H}(\tilde{\Gamma})$ or, for readability, simply $\mathcal{H}$. The set $\mathcal{H}$ describes all possible robot service rates that can be obtained by a convex combination of these basis allocations. It has been shown in [23] that $\mathcal{H}$ is the closure of Λ. Taking into account the non-zero finite transit time of robots, in fact any rate vector in the interior of $\mathcal{H}$ (i.e., any rate vector in Λ) can be served, if it is known.

A. An Illustrative Example

See figure 1, which shows the capacity region when $K = 2$, $N = 3$. The labels such as (x, y) are given to the basis allocations on the Pareto boundary to denote that they can be achieved by allocating an integer number of robots x to flow 1 and y to flow 2. Note in particular that the point (R_{max}, R_{max}) is outside the region because the only way to serve that rate is to allocate two robots full time to each of the two flows, and we have only 3 robots. The vertices on the boundary of the region, which represent basis allocations, are all in the set $\tilde{\Gamma}$.

B. Coarse-Grained Backpressure Control

The above discussion shows the capacity region for the message ferrying problem and argues that any arrival rate vector in this region can be stably served by a centralized allocation of robots if the rate vector is known. We also presented in [23] a coarse-grained backpressure control scheme called CBMF for queue-based rate-unaware scheduling, however, we found that it was neither throughput optimal nor delay efficient. In the following we present our new contribution, a fine-grained backpressure scheme which is better in both regards.

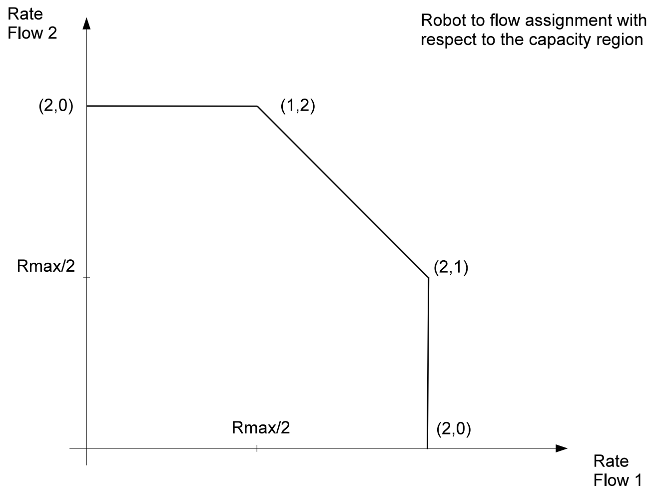

Fig. 1. Capacity region for a problem with 3 robots and 2 flows

V. Fine-Grained BackPressure Control

Let $Q_i^{src}(t)$ represent the queue size for source i, and let $Q_j^i(t)$ represent the queue size for the i^{th} flow at robot j. Let $R_{ij}^{src}(t)$ represent the rate between robot j and source i, and $R_{ij}^{sink}(t)$ represent the rate between robot j and sink i. We define link weights between each robot and each source or sink as follows:

$$w_{ij}^{src}(t) = (Q_i^{src} - Q_j^i(t))R_{ij}^{src}(t)$$
$$w_{ij}^{sink}(t) = (Q_j^i(t))R_{ij}^{sink}(t)$$

Consider the weighted Bipartite graph consisting of all robots on one side and all sources and sinks on the other, with edges from each robot to each source and sink, with edges labelled with the above weights. The maximum weight matching of robots to sources/sinks can be computed in polynomial time (such as via the Hungarian algorithm).

The Fine-Grained Backpressure-based message ferrying algorithm (FBMF) can be described quite simply as follows: at each time slot perform the maximum weight matching. If a robot i is matched to a particular source or sink, to a neighboring node on the graph that is one step closer to the matched source/sink. If it is already at the position of that source/sink, keep it there. If it is matched to a source, schedule packets to be moved from that source to the the corresponding queue of the robot. If it is matched to a sink, schedule packets to be moved from the robot to that sink.

Although it is inspired by traditional Backpressure scheduling for wireless networks, the standard proof of throughput optimality cannot be directly applied to derive or analyze FBMF. This is because, unlike the traditional formulations of Backpressure, the link rate is no longer independent of the action (which consists not only of link activations but also position change of robots, affecting their rates to all source / sink pairs). For the simplest setting of a single robot, and a single source-sink pair, however, its queue and position dynamics can be analyzed in closed form and used to prove throughput optimality.

VI. Single-Flow Single-Robot Analysis of FBMF

The following Theorem establishes throughput optimality for the simplest setting of a single robot, and a single source-sink pair.

***Theorem** 2:* Given a single source-sink pair and a single robot, assuming constant arrival of data to the source with rate $\lambda < R_{\max}/2$, FBMF ensures stability.

We will prove this theorem in two steps. First we show that when queues are sufficiently loaded the robot will make a full cycle (from sink to source and back). Then, we show that so long as the arrival rate is within the capacity region, over each cycle the change in the total queue size of the system (the sum of the source and robot queue values) is negative for sufficiently large queues.

A. Limit Cycle

***Proposition** 1:* The robot will execute a full cycle moving to a source, then the sink, then the source again, if the source queue is sufficiently loaded initially.

Proof of Proposition 1: Say the robot is at position $k \in [0, 1, 2, \ldots d]$, where 0 is the source node and d is the sink node location. Let $R(k)$ represent the rate function (which is a decreasing monotone function of distance) with respect to the source at this position, and $R(d-k)$ represent the rate function with respect to the sink at this position.

Consider the line l_k that is defined by the equation:

$$(Q_1 - Q_2)R(k) = Q_2R(d-k)$$

For notational convenience, let $y = l_k(x)$ be the functional representation of this line.

By definition of the controller, if above (or at) this line the robot moves to the sink and below this line the robot moves towards the source.

Define now the regions $\rho_k \,\forall\, k \in [1, \ldots d]$ as the set of queue pairs $\{(Q_1, Q_2)\}$ such that:

$$Q2 \geq l_k(Q_1) \text{ AND } Q2 < l_{k-1}(Q_1),$$

denote ρ_0 as the set $\{(Q_1, Q_2)\text{such that } Q2 \geq l_0(Q_1)$ and ρ_{d+1} as $\{(Q_1, Q_2)\text{such that } Q2 < l_d(Q_1)\}$.

These lines and regions are illustrated in figure 2, for the case when $d = 8$.

***Lemma** 1:* $\forall \rho_i \in [\rho_0, \ldots \rho_d]$, if current robot position $k \geq i$ and $(Q_1(t), Q_2(t)) \in \rho_i$, the robot will move towards the sink. If it is already at the sink, it will stay there until the queue conditions change to cause a movement to the source.

Proof: Since the queues are $Q_1, Q_2 \in \rho_i$, it follows that:

$$(Q_1 - Q_2)R(i) \leq Q_2R(d-i)$$
$$(Q_1 - Q_2)R(i-1) > Q_2R(d-i+1)$$

Recall that the robot is in position $k \geq i$, it follows that:

$$(Q_1 - Q_2)R(k) \leq (Q_1 - Q_2)R(i)$$
$$Q_2R(d-k) \geq Q_2R(d-i)$$

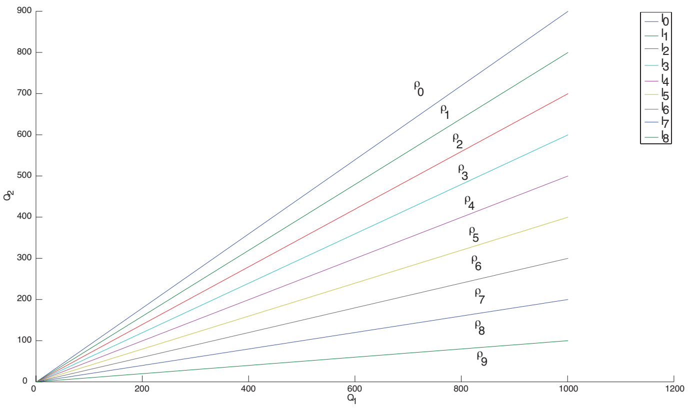

Fig. 2. Linear regions defining movement direction for the robot

This implies that:

$$(Q_1 - Q_2)R(k) \leq Q_2 R(d-k)$$

thus the robot will be moving toward the sink. ■

***Lemma** 2:* $\forall \rho_i \in [\rho_1 \ldots \rho_{d+1}]$ if the current robot position $k < i$ and $(Q_1(t), Q_2(t)) \in \rho_i$, the robot will move towards the source. If it is already at the source, it will stay there until the queue conditions change to cause a movement to the sink.

Proof: Since the queues are $Q_1, Q_2 \in \rho_i$, it follows that:

$$(Q_1 - Q_2)R(i) \leq Q_2 R(d-i)$$
$$(Q_1 - Q_2)R(i-1) > Q_2 R(d-i+1)$$

Recall that the robot is in position $k < i$, which can be written as $k \leq i-1$ it follows that:

$$(Q_1 - Q_2)R(k) \geq (Q_1 - Q_2)R(i-1)$$
$$Q_2 R(d-k) \leq Q_2 R(d-i+1)$$

This implies that:

$$(Q_1 - Q_2)R(k) \geq Q_2 R(d-k)$$

thus the robot will be moving toward the source. ■

Now, it is easy to see that if at each time t, if it is true that the robot's queue sizes do not evolve so drastically that their values at the next time step cross two of these lines, the robot will move all the way from the source to the sink, and vice versa (i.e., it will not turn back).

Consider that robot is currently at position k and region ρ_i, for some $i < d$.

Case 1: $k \geq i$

$$Q_2(n) > l_i(Q_1(n)) \tag{1}$$

This in turn implies that

$$Q_2(n) > \frac{R(i)}{R(i) + R(d-i)} Q_1(n) \tag{2}$$

By Lemma 1, it will move towards the sink.

Assuming unit time steps, the two queues will evolve as follows:

$$Q_2(n+1) = Q_2(n) - R(d-k)$$
$$Q_1(n+1) = Q_1(n) + \lambda$$

From the above, we will have that

$$Q_2(n+1) > \frac{R(i)}{R(i) + R(d-i)} Q_1(n) - R(d-k)$$

and the new position will be $k+1$.

We now wish to make sure that the new queue pair $(Q_1(n+1), Q_2(n+1))$ should not be in region ρ_{i+2}.

In order to achieve that, we wish to ensure that $Q_2(n+1) < l_{i+2}(Q_1(n+1))$ as this implies that:

$$(Q_1(n+1) - Q_2(n+1))R(i+2) < Q_2(n+1)R(d-i-2)$$

This can be written as:

$$Q_2(n+1) > \frac{R(i+2)}{R(i+2) + R(d-i-2)} Q_1(n+1)$$
$$> H(i+2) Q_1(n+1)$$

with $H(x)$ defines as:

$$H(x) = \frac{R(x)}{R(x) + R(d-x)}$$

By substituting with respect to the queues at time n we obtain:

$$Q_2(n) - R(d-k) > H(i+2)\, Q_1(n) + \lambda \tag{3}$$

At this point, by recalling the inequality on eq. (2), we have:

$$Q_2(n) - R(d-k) > Q_1(n) - R(d-k)$$
$$> H(i+2)\, Q_1(n) + \lambda \Delta t$$

Thus we can impose a constraint on the $Q_1(n)$ so that the eq. (3) is enforced, that is:

$$Q_1(n) - R(d-k) > H(i+2)\, Q_1(n) + \lambda$$
$$\Rightarrow Q_1(n) > \frac{\lambda + R(d-k)}{1 - H(i+2)}$$

The above analysis can be extended to all $i, k \geq i$ to yield a queue size Q_1 such that the robot will always move to the sink, so long as Q_2 satisfies the relationship (1).

Case 2: $k < i$

Repeating the above arguments, using Lemma 2 will yield another large enough Q_1 such that the robot will always move to the source.

Putting these two together, we have that there is a queue $Q_1(n)$ large enough (can take the bigger of the two cases) that the robot will, no matter its initial position, in finite time find its way to the sink (possibly after first visiting the source). It will then monotonically move from the sink to the source and from the source to the sink, completing a full cycle.

This concludes the proof of Proposition 1. □

B. Negative Drift Over Cycle

***Proposition** 2:* If the robot moves cyclically between the source and sink, there exists a lower bound on queue-sizes such that the change in the total queue size over a full cycle (referred to as the queue drift) is negative.

Proof of Proposition 2: Let us consider a scenario where there is a pair of sink/sources and a robot moving back and forth. Let us denote with Q_1 and Q_2 the queue of the source and robot, respectively. Furthermore, let us denote with λ the arrival rate, R and r_m are the maximum and minimum throughput, respectively. Finally, let us denote with x and y the time spent by the robot at the source and sink, respectively, and let d be the time taken to move directly between the two during which time the robot obtains an average throughput of r_a with respect to whichever node it is moving towards.

Let us suppose the robot to be at the sink and ready to head back to the source at time t_1. Successively, let us denote by t_2 the robot is done collecting data from the source and ready to head back to the sink and finally the time t_3 when it is done delivering data to the sink and ready to head back to the source again. We investigate under what conditions the total queue drift is ensured to be negative over a full round trip, that is:

$$\Delta Q(t_1,t_3) = \Big(Q_1(t_3)+Q_2(t_3)\Big) - \Big(Q_1(t_1)+Q_2(t_1)\Big) < 0$$

The algorithm FBMF ensures that the following inequalities hold:

$$\begin{aligned}
\Big(Q_1(t_1)-Q_2(t_1)\Big)\, r_m &> Q_2(t_1)\, R \\
\Big(Q_1(t_2)-Q_2(t_2)\Big)\, R &< Q_2(t_2)\, r_m \\
\Big(Q_1(t_3)-Q_2(t_3)\Big)\, r_m &> Q_2(t_3)\, R
\end{aligned}$$

The queues at the different time steps can be worked out as follows:

$$\begin{aligned}
Q_1(t_2) &= Q_1(t_1) + (\lambda - R)\, x + d\,(\lambda - r_a) \\
Q_2(t_2) &= Q_2(t_1) + R\, x + d\, r_a \\
Q_1(t_3) &= Q_1(t_2) + \lambda\,(y+d) \\
Q_2(t_3) &= Q_2(t_2) - R\, y - d\, r_a
\end{aligned}$$

In the following, for simplicity, we drop the common time index t_1 on Q_1 and Q_2 unless it is needed for clarity. In particular, the following hold:

I :
$$Q_1 > \frac{R+r_m}{r_m} Q_2$$

II :
$$Q_1 < \frac{r_m+R}{R} Q_2 + r_m\, x - \lambda\, x + 2\, R\, x + dr_a(\frac{r_m}{R}+1) - \lambda d$$

III :
$$Q_1 > \frac{R+r_m}{r_m} Q_2 + \frac{R^2}{r_m}(x-y) - \lambda(x+y) + R\,(2\,x-y) - 2\lambda d + dr_a$$

Let us now include also as a fourth inequality the condition by which the total drift is negative, that is:

$$\Big(Q_1(t_3)+Q_2(t_3)\Big) - \Big(Q_1(t_1)+Q_2(t_1)\Big) < 0$$

Algebraic manipulations based on the definition of queues at each time yield that this is equivalent to:

IV :
$$(R-\lambda)\, y - \lambda\, x + d\,(r_a - 2\,\lambda) > 0$$

Let us now denote the following constant quantities:

$$\begin{aligned}
\alpha &= \frac{R+r_m}{r_m} \\
\beta &= \frac{r_m+R}{R} \\
\gamma &= (r_m - \lambda + 2\, R) \\
\eta &= \left(\frac{R^2}{r_m} + 2\, R - \lambda\right) \\
\chi &= \left(\frac{R^2}{r_m} + \lambda + R\right) \\
\phi &= (R-\lambda) \\
b_1 &= dr_a(\frac{r_m}{R}+1) - \lambda d \\
b_2 &= -2\lambda d + dr_a
\end{aligned}$$

The above four inequalities (**I** through **IV**) can then be expressed in a simpler form as follows:

$$\begin{aligned}
Q_1 &> \alpha Q_2 \\
\gamma\, x &> Q_1 - \beta Q_2 + b_1 \\
\chi y &> \eta\, x + \alpha Q_2 - Q_1 + b_2 \\
\phi\, y &> \lambda\, x - b_2
\end{aligned}$$

It can be noticed that we have 4 inequalities and 4 unknowns, namely Q_1, Q_2, x, y. The first inequality implies that:

$$Q_2 < \frac{Q_1}{\alpha}$$

Taking this into account in the second inequality, we have:

$$x > \frac{1}{\gamma}\left[Q_1\left(1-\frac{\beta}{\alpha}\right) + b_1\right] \tag{4}$$

Let us now work out the third and fourth inequalities:

$$\begin{aligned}
y &> \frac{\eta\, x + b_2}{\chi} \\
y &> \frac{\lambda\, x - b_2}{\phi}
\end{aligned} \tag{5}$$

Now we want impose a condition on x such that if the third condition holds the fourth is implied, that is:

$$\frac{\eta\, x + b_2}{\chi} > \frac{\lambda\, x - b_2}{\phi}$$

From which it follows that:

$$x > \frac{-b_2\,(\phi+\chi)}{\phi\,\eta - \lambda\,\chi} \tag{6}$$

We now show that:

$$\phi\,\eta - \lambda\,\chi > 0 \quad \iff \quad \lambda < R/2$$

The left term can be rewritten as:

$$\frac{\eta}{\chi} > \frac{\lambda}{\phi}$$
$$\Rightarrow \frac{R^2/r_m + 2R - \lambda}{R^2/r_m + \lambda + R} > \frac{\lambda}{R - \lambda}$$

Straightforward algebraic manipulations yield that the above is equivalent to the arrival rate being less than the maximum possible throughput: $\lambda < R/2$, which is the right term.

Let us now go back to the inequality (6). It should be noticed that if $b_2 > 0$ the right hand side would be negative making the inequality trivially satisfied (since the duration x is non-negative). In particular, $b_2 > 0$ implies that $\lambda < \frac{r_a}{2}$, which in turn implies that there is no loss of efficiency in transfer due to transit.

In this case, the queue drift is negative if the inequalities (4) and (5) hold. Note that for any value of $Q_1 > 0$ such a x and y to satisfy these inqualities can be easily determined. In particular, the controller will pick the smallest x such that

$$x > \frac{1}{\gamma}\left[Q_1 - \beta Q_2 + b_1\right],$$

similarly it will pick the smallest y such that

$$y > \frac{\eta\,x + \alpha\,Q_2 - Q_1 + b_2}{\chi},$$

which will obviously satisfy the above condition.

Let us next consider the case in which $b_2 < 0$ which implies $\lambda > \frac{r_a}{2}$. This now implies that the condition in (6) is not trivially satisfied any longer, but rather it depends upon the value of Q_1. As Q_1 is increased the second inequality holds for larger and larger x, eventually, for a large enough Q_1, it will be large enough to imply the above condition. Just as in the above case, also in this case the controller will pick the smallest x such that

$$x > \frac{1}{\gamma}\left[q - \beta Q_2 + b_1\right],$$

similarly it will pick the smallest y such that

$$y > \frac{\eta\,x + \alpha\,Q_2 - Q_1 + b_2}{\chi},$$

which will satisfy all the inequalities, resulting in negative drift over the cycle. This concludes the proof of Proposition 2. □

Proof of Theorem 2: Together, Propositions 1 and 2 imply that so long as the arrival rates in the throughput region, for all source queue values beyond a certain bound, the total queue drift over a cycle is always negative. Note that with FBMF there is no way for the robot queue to grow unboundedly while keeping the source queue bounded, since for a sufficiently high difference between the robot and source queues, it will communicate with the sink long enough to bring that difference to within a constant; thus the above statement is equivalent to stating that for all total queue values beyond a certain bound the drift will be negative. This boundedness criterion ensures stability. □

VII. Simulations

We now turn to simulation results to evaluate the performance of FBMF. In our simulations, for the single source/single sink setting, we assume that sources and sinks are placed in a linear graph at locations 1 and 10. The rate as a function of distance is chosen to be

$$R = \log\left(1 + \frac{c}{1 + d^{\eta}}\right),\ \eta = 2,$$

so that $R_{\max} = \log(1 + c)$.

A. 1 Robot - 1 Flow

Figure 3 shows how the source and robot queues evolves when the arrival rate is within the maximum throughput of $R_{\max}/2$. In this case, the queues are initially overloaded, and the robot spends sufficient time at the sources and sinks and cycles between them to bring down the load. Eventually it settles down to a cyclic pattern going between the source and sink and spending just enough time at both ends to make the total queue change over a cycle to be 0.

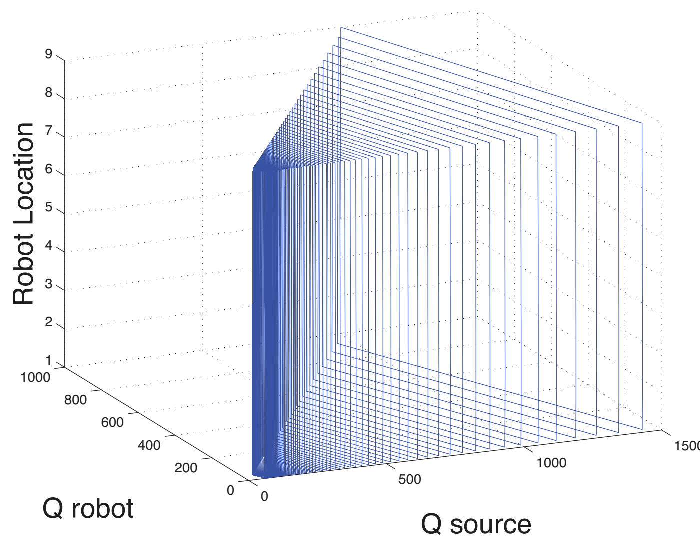

Fig. 3. 1 Robot - 1 Flow - Stable arrival rate within 10% of maximum throughput

If the arrival rate is higher than the throughput, then, as shown in Figure 4, despite the robot's best efforts of going back and forth between the source and sink, the queue sizes spiral outwards increasing without bound, due to instability.

For this case, we also compare with the performance of the CBMF algorithm proposed in [23]. Figure 5 shows how the sum of the queues (robot+source) varies with time for different arrival rates (expressed as percentage of maximum possible throughput $R_{\max}/2$) for the CBMF algorithm for a fixed epoch size of 500 iterations.

It can be seen that for this setting, the system is not stable when arrival rates are sufficiently high (99 %) revealing that CBMF is not throughput-optimal. On the other FBMF is stable at all these rates. CBMF can be stabilized for higher rates by increasing epoch size, but this has an adverse effect on the average queue sizes (and hence on the average delay) for lower

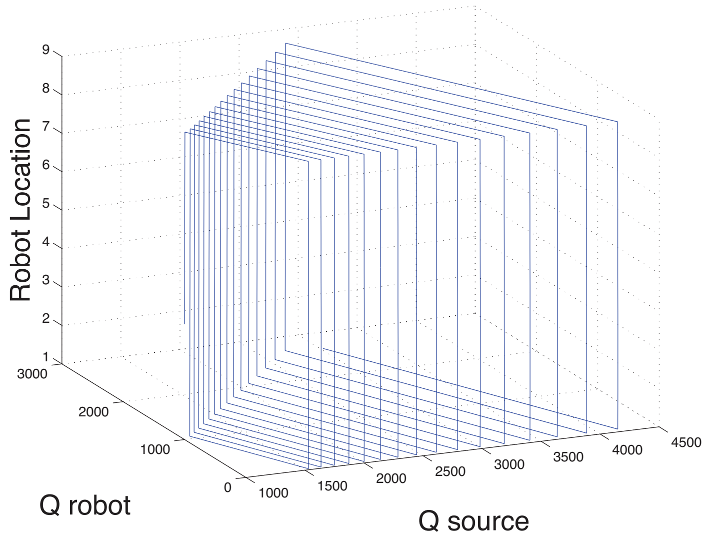

Fig. 4. 1 Robot - 1 Flow - Unstable arrival rate exceeding maximum throughput by 10%

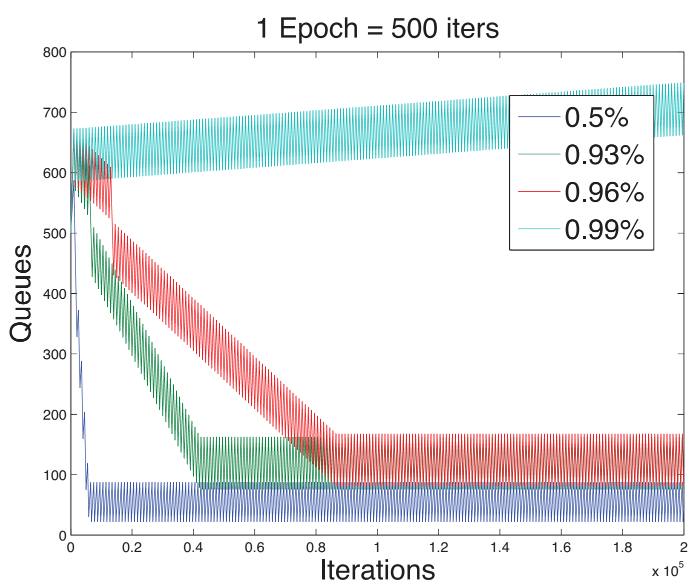

Fig. 5. 1 Robot - 1 Flow - CBMF - 1 Epoch = 500 Iterations

	50%	93%	96%	99%
CBMF $T = 500$	296	368	372	∞
CBFM $T = 5000$	2921	3638	3689	3738
FBMF	52	159	281	2160

TABLE I. AVERAGE END-TO-END DELAY OF PACKETS FROM SOURCE TO SINK COMPARISON. ARRIVAL RATES EXPRESSED AS PERCENTAGE OF MAXIMUM THROUGHPUT.

rates. This is shown in Table I which compares the average end-to-end delay of packets from source to sink (in units of time) for different arrival rates for CBMF with two different epoch sizes, and FBMF. As noted, for CBMF with $T = 500$, the algorithm is unstable (infinite average delay) at 99% of the maximum throughput. With $T = 5000$ it is now stable, but in this case the average delay is extremely high at all rates. FBMF offers both throughput optimality (provably in this case) as well as relatively low average delay at all rates, without any

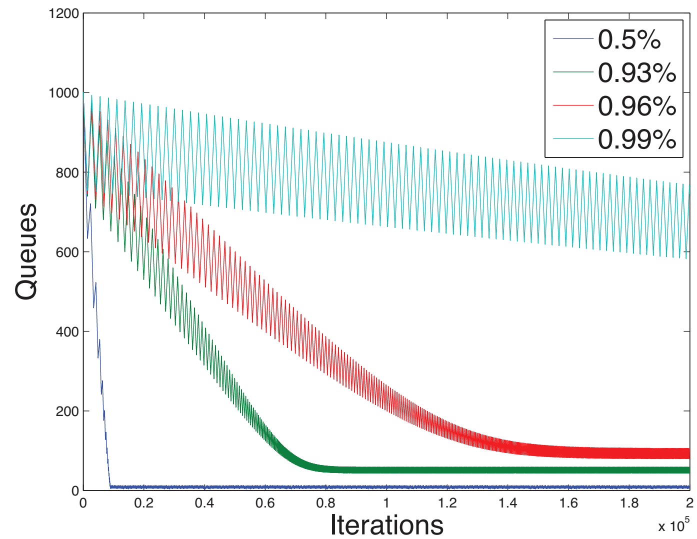

Fig. 6. 1 Robot - 1 Flow - FBMF

parameter setting.

B. 2 Robots - 1 Flow

We illustrate what happens with 2 robots and 1 flow. We find that particularly as arrival rates get close to the theoretical limit ($R_{\max}$ in this case), the movements of the two robots start to mirror each other automatically, that is when one reaches the source, the other reaches the sink, and vice versa. This is shown in Figure 7. Even for arrival rates very close to the maximum rate we observe that the queues remain stable, as seen in figure 8.

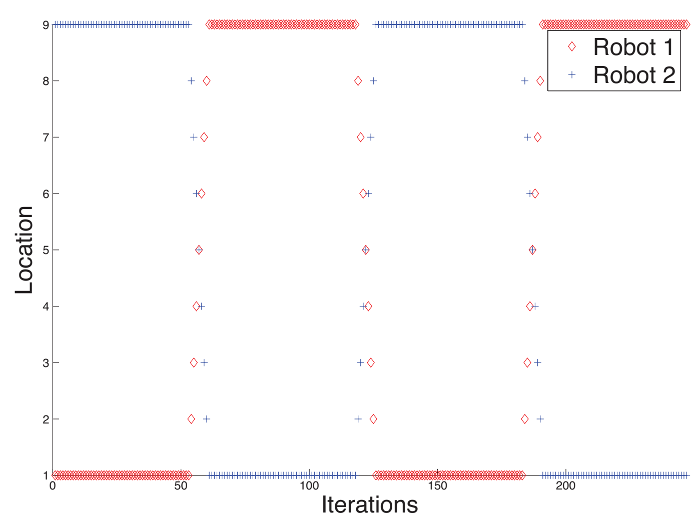

Fig. 7. 2 Robot - 1 Flow - Stable

VIII. CONCLUSIONS

In this work, we addressed the motion control problem for a set of robots which move to ferry messages between a given set of statically-places nodes. A novel fine-grained backpressure message ferrying algorithm has been proposed. A

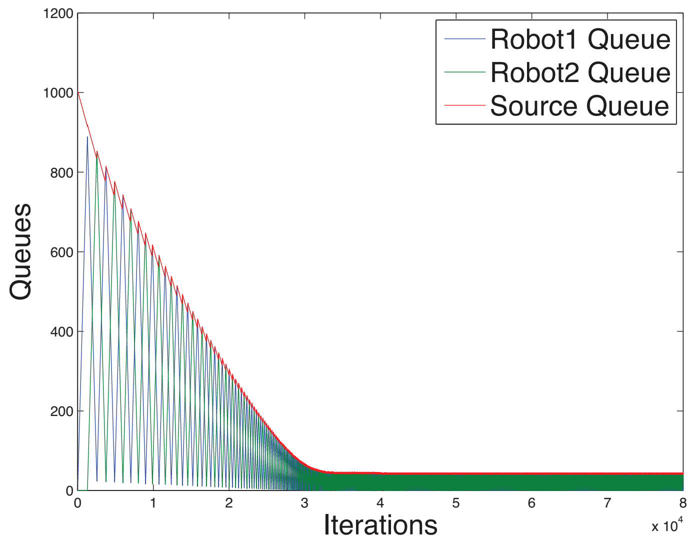

Fig. 8. 2 Robot - 1 Flow - Stable

theoretical analysis of throughput optimality has been provided for the simplest setting, that is single-flow, single-robot, deterministic arrival. Numerical results indicate that the algorithm outperforms our prior work on a coarse-grained algorithm for this problem. The design and analysis of a throughput optimal policy for the general case is posed as a challenging open problem for network theory. Finally, in this work we have assumed an interference free scheduling is possible (for example, via frequency/time/code allocation) In future work we plan to also consider interference between communications involving different robots.

Acknowledgment

This work has been supported in part by NSF under award number 1217260.

References

[1] L. Tassiulas and A. Ephremides, “Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks”, IEEE Tran. Automatic Control, 37(12), pp. 19361949, 1992.

[2] M. Neely, E. Modiano, and C. Rohrs, “Dynamic power allocation and routing for time varying wireless networks,” *IEEE Conf. Computer Communications (INFOCOM),* 2003.

[3] A. Eryilmaz and R. Srikant, “Fair resource allocation in wireless networks using queue-Length-Based scheduling and congestion control,” *IEEE Conf. Computer Communications (INFOCOM),* 2005.

[4] X. Lin and N. Shroff, “Joint rate control and scheduling in multihop wireless networks,” *IEEE Conf. Decision Control (CDC),* 2004.

[5] M. J. Neely. Stochastic Network Optimization with Application to Communication and Queueing Systems. Morgan and Claypool, 2010.

[6] L. Bui, R. Srikant, A.L. Stolyar, Optimal Resource Allocation for Multicast Flows in Multihop Wireless Networks , Philosophical Transactions of the Royal Society A, 2008, Vol. 366, pp.2059-2074.

[7] L. Bui, R. Srikant, A.L. Stolyar, A Novel Architecture for Delay Reduction in Back-pressure Scheduling Algorithm , IEEE/ACM Trans. on Networking, 2011, Vol.19, No.6, pp.1597-1609.

[8] Dongyue Xue, Robert Murawski, and Eylem Ekici, Distributed Utility-Optimal Scheduling with Finite Buffers, Proceedings of WiOpt 2012, Paderborn, Germany, May 2012.

[9] B. Ji, C. Joo, and N. B. Shroff, “Delay-Based Back-Pressure Scheduling in Multi-Hop Wireless Networks,” *IEEE INFOCOM*, Shanghai China, April 2011.

[10] L. Huang, S. , M. J. Neely and B. Krishnamachari, “LIFO-Backpressure Achieves Near Optimal Utility-Delay Tradeoff, *IEEE/ACM Trans. on Networking*, 2013.

[11] J. Ryu, L. Ying and S. Shakkottai, “Back-pressure Routing for Intermittently Connected Networks, Proceedings of IEEE Infocom mini-conference, San Diego, CA, 2010.

[12] Majed Alresaini, Maheswaran Sathiamoorthy, Bhaskar Krishnamachari and Michael J. Neely, “Backpressure with Adaptive Redundancy (BWAR)”, *IEEE International Conference on Computer Communications (IEEE INFOCOM)*, Orlando, FL, USA, March, 2012.

[13] L. Jiang and J. Walrand, “Approaching Throughput-Optimality in Distributed CSMA Scheduling Algorithms with Collisions,” IEEE/ACM Transactions on Networking, vol. 19, no. 3, pp. 816 829, Jun. 2011.

[14] S. Moeller, A. Sridharan, B. Krishnamachari, O. Gnawali, “Routing Without Routes: The Backpressure Collection Protocol,” *ACM/IEEE IPSN*, April, 2010.

[15] R. Laufer, T. Salonidis, H. Lundgren, P. Le Guyadec, “Design and Implementation of Backpressure Scheduling in Wireless Multi-Hop Networks: From Theory to Practice,” *ACM Mobicom*, 2010.

[16] J. Ryu, V. Bhargava, N. Paine, and S. Shakkottai, “Back-pressure routing and rate control for icns, ACM Mobicom, 2010.

[17] G. Wang, G. Cao, P. Berman, and T. La Porta, “Bidding protocols for deploying mobile sensors,” *Mobile Computing, IEEE Transactions on*, vol. 6, no. 5, pp. 563–576, 2007.

[18] A. Gasparri, B. Krishnamachari, and G. S. Sukhatme, “A framework for multi-robot node coverage in sensor networks,” *Annals of Mathematics and Artificial Intelligence*, vol. 52, no. 2-4, pp. 281–305, Apr. 2008.

[19] M. M. Zavlanos and G. J. Pappas, “Distributed connectivity control of mobile networks,” *Trans. Rob.*, vol. 24, no. 6, pp. 1416–1428, Dec. 2008.

[20] M. M. Zavlanos, A. Ribeiro, G. J. Pappas, “Network Integrity in Mobile Robotic Networks,” Automatic Control, IEEE Transactions on , vol.58, no.1, pp.3,18, Jan. 2013

[21] D. K. Goldenberg, J. Lin, S. A. Morse, B. E. Rosen, and R. Y. Yang, ”Towards mobility as a network control primitive, in Proc. 5th ACM Int. Symp. MobiHoc, 2004.

[22] Y. Yan and Y. Mostofi, ”Robotic Router Formation in Realistic Communication Environments,” IEEE Transactions on Robotics, vol. 28, no. 4, pp. 810-827, August 2012.

[23] S. Wang, A. Gasparri, B. Krishnamachari, “Robotic Message Ferrying for Wireless Networks using Coarse-Grained Backpressure Control,” short paper in IEEE Globecom 2013 Workshop - Wireless Networking and Control for Unmanned Autonomous Vehicles.