On Using Mobile Robotic Relays for Adaptive Communication in Search and Rescue Missions

Eduardo Feo Flushing[1], Luca M. Gambardella[1], Gianni A. Di Caro[2]

Abstract— **This work addresses the problem of deploying and controlling mobile relays in robotic networks. We consider a general data collection scenario in which a group of robots stream data towards one or more base stations. The robots form a wireless mobile ad hoc network relaying data towards the base stations in a multi-hop fashion. Additional robots are used as mobile relays with the aim of optimizing overall network performance in terms of data throughput. To dynamically control the positions of these relays, we use the combination of a mathematical programming model that includes a number of constraints and penalties to closely model the wireless environment, and heuristics to account for variations in wireless links stability. We demonstrate that the approach is computationally affordable for reasonably sized networks. The approach is evaluated in realistic simulation scenarios, studying the impact of the model parameters on the quality of the provided solutions. An experimental validation is also carried out in a real testbed, confirming the efficacy of the proposed approach for adapting relays' positions to the changes in the robot network.**

I. Introduction

The presence of a *reliable communication network* is a core component for the effective functioning of a *mobile multi-robot system*. When the system is deployed over large remote areas for emergency response, or is used for search and rescue in post-disaster scenarios, communications may be needed for: implementing effective distributed coordination and cooperation among the robots, streaming sensor data to a processing center, allow the real-time monitoring and teleoperation of the system from a control authority.

The provisioning of networking can be realized in a number of different ways. In general search and rescue scenarios, that are the application focus of this work, it is likely that no external infrastructure is available. To overcome this drawback, the robots are equipped with a range-limited wireless interface, forming a *mobile ad hoc network* in which communications happens in *multi-hop* modality.

A number of different approaches have been proposed in the literature to counterbalance the intrinsic limitations related to multi-hop wireless communications, ranging from the design of smart strategies for contention and sharing of the medium, to the use of adaptive routing algorithms, to the use of cooperative MIMO solutions. A strategy which is traversal to all these different approaches consists in the use of dedicated nodes specifically acting as *communication relays*. In this work we consider the use of *mobile robotic relays* and focus on how to control their positions to adapt to the dynamic changes in the topology of the robot network.

Our reference scenario is a data collection one typical of search and rescue missions: a set of *mission robots* move in the environment according to their own tasks and objectives, and need to regularly stream data (e.g., images) to one or more control centers (base stations) and can also receive data from them (e.g., for teleoperation).

We propose the use of a dual-network system: one *control network* (single-hop broadcast, long-range, low bandwidth, low QoS) and one *data network* (multi-hop, short-range, high- bandwidth, medium QoS) work in parallel. We assume that the control network is connected. Yet, due to the low capacity of the control network, robots must use the multi-hop data network to exchange most of the data with the control centers. To support effective multi-hop communications in this network, a small set of mobile robot relays is deployed specifically for this purpose.

Based on this scenario, we propose a model-based approach for *dynamically controlling the position of the mobile relays* and computing at the same time *routing paths* for multi-hop data transmissions over the data network. The general goal is to *maximize the total throughput* of the data sent from the robots to the base stations. With the aim of providing *optimality guarantees* given the life-critical task of the system, relays' positions and data routing paths are computed in a *centralized* manner. Iteratively, a *central controller*: (i) gathers position (e.g., from GPS) and traffic information from the robots using the control network, (ii) feeds this information into a mathematical model, a *mixed integer linear program* (MILP), that jointly computes plans for the relay robots and routing paths for the data to be sent to the base stations, and (iii) broadcasts the resulting output to the robots, that implement the received directives. The MILP explicitly takes into account and models effects of interference and congestion in wireless environments. Robots' positional information is used to estimate future network configurations and to derive estimates for the stability and reliability of the existing and the prospective wireless links that are going to be included in the model.

The *main contributions of the work* are:

- a mathematical framework for the dynamic planning of the positions of mobile relays based on the explicitly modeling of link stability, wireless interference, and future system configurations;

[1] Eduardo Feo Flushing and Luca M. Gambardella are with the Dalle Molle Institute for Artificial Intelligence (IDSIA), Lugano, Switzerland, {eduardo,luca}@idsia.ch

[2] Gianni Di Caro is with the Department of Computer Science, Carnegie Mellon University (CMU), gdicaro@cmu.edu.

This work was partially supported by the Swiss National Science Foundation (SNSF) through the National Centre of Competence in Research (NCCR) Robotics.

- joint computation and optimization of the plans for the relays and of the optimal paths for data routing;
- a system architecture based on the concurrent use of two networks with complementary characteristics and application, one for centralized I/O control and one for multi-hop data transmissions;
- a sensitivity analysis based on simulation and the validation of the system in a mobile, multi-robot testbed.

II. Related work

The problem of providing and maintaining ad hoc communications in a team of mobile agents is addressed in a number of domains including, among others, multi-robot exploration [1], path planning and navigation [2], task allocation and planning [3], [4], and pursuit and evasion [5].

One way to address the problem is to consider the provisioning of communications as integral part of the team mission. In this way, the agents simultaneously play the role of communication providers and mission executors. Most of these approaches enforce the *continual satisfaction of hard communication constraints* (e.g., in terms of proximity among the agents). Examples of these works include establishing permanent communication paths between a base station and a group of agents [6], [2], [7], [8], [4].

Common to most of these approaches is the inclusion of the networking constraints directly into the problem formulation. Although the resulting problem may still be computationally affordable, there are many situations in which this approach to support communications is not feasible or not really reliable. For instance, this is the case when some of the agents are not fully controllable, such as humans or dogs [9].

In principle, supporting communications does not require adding complex and costly hardware. This observation, together with the increasing availability of mobile platforms, both ground and aerial ones, has promoted the inclusion of an additional group of agents whose only objective is to enable data communications, which is the also the way we deal with the problem. Examples of this way of proceeding include the use of mobile robotic routers for building and maintaining a communication infrastructure [10], [11], [12], [13], [14], the deployment of relay nodes to enhance the network response [15], and the use of *data mules* [16] to collect data from a set of *static* nodes, and deliver these data to the end-user (e.g., a base station).

In our approach, we make use of a centralized approach for adaptively computing the path plans for the mobile relays (and the data routing paths). Compared to distributed or decentralized approaches for connectivity maintenance [17], [10], [3], [18], [19], [20], [21], [22], [23], that are indeed very popular, centralization allows to place much of the computational load to the central controller, therefore reducing the cost and the complexity of the individual team agents, which makes the whole system relatively easier to implement. Moreover, given that the central controller has an accurate, global view of the mission, it is possible to compute optimal solutions. In our case this is made possible in practice also by the fact that the solution of the MILP model is fast for reasonably sized teams.

Another aspect that justifies the choice for a centralized controller, is the observation that many classes of real-world problems are inherently centralized. The case of search and rescue mission is exemplary: given that these are life-critical missions, it is necessary to gather all the accessible information and process it globally, at a base station, that can be located in a safe place, away from mission hazards [1].

The majority of the centralized approaches for communication support in multi-robot teams focus on the maintenance of the group connectivity [8]. It is common practice to define a continuous trajectory for mobile relay nodes (or robotic routers [24]) with the objective of maximizing some network performance metric. For this reason, most of these works adopt a low-level reactive control approach, such as gradient-based controllers [8], [2], [14], and potential fields [11]. Due to their reactive nature, these approaches exhibit a myopic behavior, as they do not consider the effect of temporarily breaking connectivity in favor of higher quality network configurations later on.

In this work we follow a different strategy to exploit the use of robotic relays. We cast the problem as the one of finding the optimal placement of relay nodes inside the operation area with the goal of improving network performance. In this way, we can consider a wide range of possible network configurations that can be formed in practice by instructing the relays to move to specific locations.

The literature on the topic of relay node placement in wireless networks is wide. A vast body of literature is dedicated to topology management problems in sensor networks [25]. In these works, the typical scenario consists of a network of stationary sensors [26]. Instead, in this work we consider a dynamic scenario where the topology of the robotic network is constantly changing.

III. Data collection scenario

A. The actors

We consider a networked system consisting of four types of situated elements: *mission robots*, *mobile robotic relays*, *data collection centers*, *control center*. Henceforth, we use the word *node* to refer to an element of any of these types.

The set $\mathcal{A}$ of mission robots consists of regular mobile robots that are engaged in some mission. They move throughout a bounded area at a maximum speed of s_A m/s. Each robot $a \in \mathcal{A}$ produces a known amount of data, measured in bytes per second. The set of data collection centers $\mathcal{B}$ are the nodes where the data produced by the robots in $\mathcal{A}$ must be gathered. For sake of simplicity, and without loss of generality, we herein assume that nodes in $\mathcal{B}$ remain stationary. The set of robotic relays $\mathcal{R}$, with typically $|\mathcal{R}| < |\mathcal{A}|$, consists of mobile controllable elements capable of navigating autonomously inside the area, with a maximum speed of $s_R \geq s_A$. They can be commanded to move and/or to stay at specific locations (i.e., waypoints). Regarding the mission robots in $\mathcal{A}$, we assume that their movement is determined by some process related to the mission and it is considered external to our system: their trajectories cannot be directly controlled or modified. The control center is the centralized station where the plans for the mobile relays and the routing paths for the data to the collection centers are iteratively computed online based on the gathering of the positional information from the robots.

B. Dual-network architecture

All the robots are equipped with *two wireless network interfaces*, one to communicate with the control center over the control network, and one to communicate with each other and the data collection centers over the data network. In the control network communications are in full broadcast (i.e., a single broadcast can reach all the nodes in the network), while communications in the data network happen in multi-hop modality based on local broadcasting.

The control network does not need high bandwidth, but has to provide long-range communications, to potentially cover the entire area where the robots move. For instance, an 802.15.4 network can be conveniently used for this purpose. Our preliminary tests precisely using XBee Pro devices show that such a technology can easily support the required control communications to/from a control center of relatively large multi-robot systems. Overall, the dual-network architecture seems to be a viable option for many practical real-world scenarios and allows to exploit global information and reasoning.

Regarding the data network, we assume that all wireless interfaces operate in the same channel, share the same medium, and the antennas are ommidirectional. The access to the wireless channel is managed by a simple, randomized CSMA MAC mechanism These assumptions are in line with the standards currently implemented by the majority of wireless hardware (e.g., IEEE 802.11, IEEE 802.15.4).

C. Routing policies for multi-hop data collection

In a multi-hop data collection scenario, at any point in time, each node n is in charge of a specific amount of data which is either generated by the same node (if $n \in \mathcal{A}$), or is relayed from other nodes in the wireless neighborhood of n. Because all data produced by nodes in $\mathcal{A}$ should be gathered at the same place, namely at any one of the data collection centers, all data can be treated equally. Hence, in the scenarios that we consider it is enough to let each node know to which other nodes the data (own and not own) should be relayed to. From the data routing perspective, these nodes are called the *next hops*.

To establish the multi-hop routing paths, we let the control center issue *routing policies*. More specifically, a routing policy for a given node i is in the form of a list of entries $(m_{1i},\ d_{1i}), \ldots, (m_{ni},\ d_{ni})$, where $0 \leq d_{ji} \leq 1$ indicates the fraction of data that i should relay to node j, and $\sum_j d_{ji} \leq 1$. In fact, in our model we assume that the data flow generated by a mobile robot can be *split over multiple paths* and arrive at multiple data collection centers.

Under some circumstances, that are quite common in a mobile ad hoc network, the network may not be able to guarantee the reception of all the data being currently generated (e.g., due to limited bandwidth capacity, presence of bottlenecks, or wireless interference). To deal with such (common) situations, in our model we also include the possibility to perform a controlled *dropping* of data packets at the nodes. In practice this is similar to operate a *flow control*: if the wireless channel at the node cannot support the transmissions of a certain amount of data, the excess data can be conveniently dropped without being placed on the channel. Clearly, while dropping data is not desirable, it can be necessary. The net result is a benefit for the network since it translates into saving energy and network resources.

In a static scenario, nodes remain stationary and the data generation rates are stationary too. When information regarding the locations of nodes and data rates is available, it is then possible to determine the optimal routing policy that supports the reception of the largest amount of data to the data collection centers. This policy would remain the optimal one as long as the environment does not change: nodes stay in place and data rates remain stationary.

Instead, in a mobile scenario and/or when traffic generation changes, a previously computed routing policy may rapidly become obsolete. One way to tackle this issue is to *periodically recompute the optimal routing policies* based upon the current situation of the network. This is precisely what our systems does: using the control network, updated positional and traffic information from the nodes is gathered at the control center, and used to compute up-to-date routing policies.

D. Mobile relays to support data transmission

While an optimal routing policy supports the formation of efficient multi-hop routing paths, an intelligent *relay placement strategy* can deploy the available robotic relays in those positions within the environment that would determine an increase in throughput. Of course, in order to select the best positions that can be used for data forwarding, the computation of the relay positions and of the routing paths must be done *jointly*, since they are *interdependent*.

To minimize the computational load while guaranteeing a satisfactory path optimization, we assume that the placement of the relays is restricted to a *numerable set of candidate locations*. At any point in time, each relay is commanded to position itself at one specific location selected among the candidate locations. Using their local navigation capabilities, the robotic relays move through the environment until reaching the selected destination point. Once a new destination is received, the relay immediately heads towards it.

Similarly to the case of routing policies, in static scenarios it is possible to determine an optimal relay placement that enables the largest network performance increase. However, in dynamic scenarios, the current relay placement can quickly loose its efficacy and be no longer exploited by the routing counterpart, since the mission robots are moving and the network topology has changed, for instance. To tackle this issue, the control center *periodically recomputes the optimal placement* based upon the current situation of the network, gathered through the control network. These placements guide the relays throughout the area with the aim of supporting the formation of *local network topologies that can be exploited by multi-hop data routing*. The framework is illustrated in Figure 1.

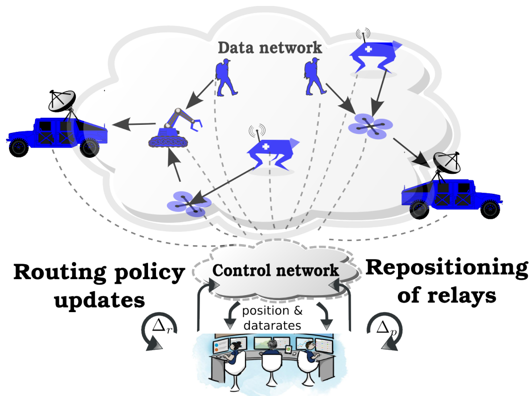

Fig. 1: The proposed framework uses a dual-network architecture and optimizes the performance of the data network through the adaptation of routing policies and relay positions.

IV. Model-based framework for optimal routing and relay placement

The main goal of this work is to adaptively and iteratively compute routing paths and relays' positions to optimize data collection in the mobile multi-hop data network.

The definition of relays' positions and of the routing paths are however interdependent problems. In fact, from the one hand, defining the most efficient routes depends upon the deployment of robotic relays in specific locations. From the other hand, an efficient deployment of robotic relays depends on the quality and the traffic of the data routes that such deployment could support. These intrinsic interdependencies force us to jointly optimize data routes and relay positions. In the following we propose an optimization approach, based upon mathematical programming, that deals with the joint problem of data routing and relay placement. We will first show the model for the static scenario, and then we will show how it is applied to the case of dynamic scenarios.

A. Mathematical formulation as a MILP

We formalize the problem by means of a MILP based on a *minimum cost flow* formulation that includes a number of additional constraints and penalties to closely model some of the main limiting factors in wireless networks.

B. Preliminaries

Let $\mathcal{R}_i^*$ be the set of candidate locations for relay $i \in \mathcal{R}$, and $\mathcal{R}^* = \bigcup_i \mathcal{R}_i^*$. Let $\mathcal{L} \subset (\mathcal{A} \cup \mathcal{B} \cup \mathcal{R}^*)^2$ be the set of communication links that can be used for data routing. When a communication link is incident to a candidate location, then the link is available iff a relay node is placed at that location. In addition, a *link cost* $\gamma : \mathcal{L} \mapsto \Re$ represents the (normalized) quality of a link, and the current network traffic load is described by $\tau : \mathcal{A} \mapsto \Re$, expressed as the data per second generated by the mobile agents. In the following, the traffic load τ, data flows, and, in general, any term related to an amount of data will be expressed in *flow units*, f_{unit}, which is simply a measure in bytes/sec.

In order to consider data drops, we include an additional *dummy node* z^* representing a virtual sink. All data that arrives to z^* is considered to be dropped. We extend our network with a set of links $\mathcal{L}^{z^*} = \{(i, z^*) \,|i \in \mathcal{A}\}$. Data traveling through link (i, z^*) represents the data that i drops (i.e., its own data or data relayed from other nodes). Each single flow unit dropped by node i is penalized by μ_i.

Shared wireless channels are necessarily *bandwidth-limited*. Exceeding the capacity of the channel can deteriorate the performance of the entire network, producing a large amount of packet loses due to the wireless interference produced. To account this issue we let $\mathcal{I}_i$ be the set of nodes that share the wireless channel with node $i \in \mathcal{A} \cup \mathcal{B} \cup \mathcal{R}^*$. Γ represents the bandwidth capacity of the network, defined in terms of flow units, that is, the available bandwidth between any pair of nodes.

The main decision variables in the model are the *flow variables* f_{ij}, which denote the amount of flow, or desired traffic load, through link (i, j), and the *binary positional variable* y_i indicates whether a relay is placed at $i \in \mathcal{R}^*$. These two sets of variables determine a solution to the problem. The routing policy is induced by the links that are used to circulate flow: $\{(i,\ j) \in \mathcal{L} \cup \mathcal{L}^{z^*} \mid f_{ij} > 0\}$. The total amount of data dropped in the network is given by $\mathcal{F}^d = \sum_{i \in \mathcal{A}} f_{iz^*}$. The MILP model is the following:

$$\min \quad \sum_{i \in \mathcal{A}} \mu_i f_{iz^*} + \sum_{(i,\ j) \in \mathcal{L}} (1 - \gamma_{ij}) f_{ij} \tag{1}$$

subject to

$$\sum_{(i,\ j) \in \mathcal{L}} f_{ij} - \sum_{(j,\ i) \in \mathcal{L}} f_{ji} = \tau_i - f_{iz^*} \qquad \forall i \in \mathcal{A} \tag{2}$$

$$\sum_{(i,\ j) \in \mathcal{L}} f_{ij} - \sum_{(j,\ i) \in \mathcal{L}} f_{ji} = 0 \qquad \forall i \in \mathcal{R}^* \tag{3}$$

$$\sum_{i \in \mathcal{B}} \sum_{(j,\ i) \in \mathcal{L}} f_{ji} = \sum_{k \in S} \tau_k - \mathcal{F}^d \tag{4}$$

$$\sum_{j \in \mathcal{I}_i} \sum_{(j,\ k) \in \mathcal{L}} f_{jk} \leq \Gamma \qquad \forall i \in \mathcal{A} \cup \mathcal{R}^* \cup \mathcal{B} \tag{5}$$

$$y_i = 1 \iff \sum_{(i,\ j) \in \mathcal{L}} f_{ji} > 0 \qquad \forall i \in \mathcal{R}^* \tag{6}$$

$$\sum_{j \in \mathcal{R}_i^*} y_j = 1 \qquad \forall i \in \mathcal{R} \tag{7}$$

$$y_i \in \{0, 1\} \qquad \forall i \in \mathcal{R}^* \tag{8}$$

$$f_{ij} \geq 0 \qquad \forall (i,\ j) \in \mathcal{L} \cup \mathcal{L}^{z^*} \tag{9}$$

We remark that the MILP can also be used to compute exclusively routing policies by fixing the relays at their current locations, that is, by letting the sets $\mathcal{R}_i^*$ contain only one element corresponding to the current location of relay i.

We also note that the MILP can be solved with standard solvers that exhibit anytime properties: solutions are progressively and monotonically improved over computation time and can be retrieved with formal error bounds on optimality. To exert a proper control of the time spent computing a solution, we can set a maximum time to the solver, after which the solver stops the optimization and returns the best solution found. In Section V we show that a standard solver can obtain near-optimal solutions in reasonably sized scenarios.

C. Dealing with mobile and cluttered environments

To cope with mobile and dynamic scenarios we propose to repeatedly solve the MILP to compute routing policies and relay placements utilizing up-to-date information about the topology of the network and the positions of the nodes. Yet there are some issues that must be adequately addressed when implementing this *iterative re-optimization approach*.

First, the rate of the re-optimization procedure should be set taking into account: (i) the time needed to compute the optimal solution to the MILP, or a sub-optimal solution of good quality; (ii) the overhead and cost induced by each recomputation and delivery of solutions; and (iii) the impact that frequent solution updates could have over the system.

We note that each of these factors behave differently in the case of routing policies and relay placements. On the one hand, frequent recomputations of routing policies are feasible and beneficial from a practical point of view because the cost and the time required to compute and implement a routing policy is relatively low (the corresponding MILP has a reduced size). On the other hand, computing a relay placement solution is a harder problem that already involves the computation of routing policies.

For this reason, we propose to decouple the routing and placement aspects during the iterative process, and interlace the recomputation of routing policies, and of placement of relays. To this end, we define two time intervals, Δ_r, and Δ_p that define the time between the recomputation of routing policies and relay placement respectively. As already mentioned, recomputing a routing policy alone amounts to solve the MILP with $\mathcal{R}_i = \{x_i\}$, where x_i denotes the position of relay i. For both routing policies and relay placements, the maximum time allocated for computing a solution to the MILP should be less than the time between recomputations.

The second issue that we take into account in our framework regards the *stability and feasibility of routing policies within the time interval Δ_r between each recomputation*. In particular we note that if some of the wireless links used to relay data in a routing policy cease to exists (e.g., because the corresponding nodes are no longer within each others' transmission range), then the overall performance of the network degrades and data losses occur.

To address this issue we propose a preselection procedure for the set of links $\mathcal{L}$ before recomputing a routing policy. We first enumerate a set of potential links based upon the knowledge of the current positions of all nodes, transmission range, and maximum speeds. Secondly, for each potential link $i \to j$ we predict the positions of i and j during the time interval before the next recomputation based on the estimated velocity vectors v_i and v_j. Based on this prediction, we derive the *expected time to live* ttl_{ij} of link $i \to j$ that estimates the fraction of time that nodes i and j would be able to communicate before the next update of routing policy. Using the values of ttl_{ij} we perform a *link pruning* procedure after which any link $i \to j \in \mathcal{L}$ with $ttl_{ij} < ttl^*$ is excluded from the set $\mathcal{L}$. In other words, we enforce the model to generate routing policies that are composed of links that have a expected time to live of at least ttl^*.

We remark that we can accommodate the presence of *obstacles* (e.g., in a cluttered environment) through the definition of $\mathcal{L}$. For instance, if the obstacles have a shielding effect on the radio signal, and we have a map of the area indicating their locations, we can determine which links intersect with any of the obstacles, and exclude them from $\mathcal{L}$. In the experimental part, we consider two scenarios featuring several obstacles.

V. Sensitivity analysis

In this section we perform a sensitivity analysis on some of the parameters used in the model for a better understanding of their relation to the performance of the system. The study is based on extensive simulations and specifically analyze the performance of the mobile network after implementing the routing policies and relay placements provided by the framework. The results will provide appropriate values for these parameters, which will be used in the experimental validation presented in the next section.

In order to perform realistic simulations, we use the integrated simulation environment *RoboNetSim* proposed in [27], which combines a multi-robot simulator (ARGoS [28]) with a realistic network simulator (NS-3 [29]). To obtain solutions to the mathematical models we use CPLEX®. We set a maximum time of 5 seconds to compute a solution, after which the solver returns the best solution found with a measure of its quality in terms of a relative optimality gap. We note that, in most of the cases, the solver is able to obtain the optimal solution or provide a near-optimal solution with a gap $\leq 5\%$ within the time limit.

The considered scenarios are based on a square area of 500×500 m^2. Base stations are statically deployed at the four corners of the area. Mission robots move with a maximum speed $s_A = 1.5$ m/s. We assume that robotic relays are more agile, and move at a speed of 7.5 m/s. These settings are inspired by the current use of aerial robots for supporting communications in the field (e.g., in search and rescue missions). The number of mission robots in these scenarios is set to 30. The number of relays is varying according to the evaluation goals. Mission robots produce data according to a random traffic generation rate from 10 KB/s up to $100 KB/s$.

We simulated 802.11g Wi-Fi networks with the transmission rate of 12 Mbps. This rate allows to obtain a transmission range of roughly 200 m, which is suitable for real-world operations in outdoor and vast areas. We use a log-distance propagation loss model with default parameters.

A. Preselection of wireless links

The first part our analysis is aimed at studying the effect of the preselection of links. As seen in Section IV-B, a necessary input to the framework is the set $\mathcal{L}$ that represents the links that can be used to send data.

A common approach to define the set of wireless links is to establish a distance threshold and to consider that any two nodes in the network can communicate (i.e., there exist a link between them) whenever they are separated by a distance less than the threshold (e.g., the euclidean disk model [30]).

The disk model is a simple and deterministic approach to define the set $\mathcal{L}$. However, setting distance threshold (also called transmission range) is not a straightforward task. Indeed, a bad choice can have serious consequences on the network. In our case, this decision directly affects the quality of the optimal routing policy. On the one hand, a conservative threshold may discard links that could be useful in practice. On the other hand, a loose threshold could introduce links that exhibit poor performance in practice. Our goal is to understand this implicit trade-off between threshold value and the performance of the network.

Discussion: Figure 2 (left) shows the effect of different distance thresholds over the performance of static networks. We consider only static scenarios, in order to focus on the stationary network performance. Results are applicable to mobile networks. As performance metric we use the *delivery ratio* of all data generated by the mission robots: the number of packets received at any of the base stations divided by the total amount of data packets generated by the mission robots. For each threshold considered we performed 20 simulation runs lasting 5 minutes each.

We observe a peak in performance for a threshold of 110 meters. After this value, performance tends to decrease. We also notice that, with the simulation settings chosen, links that have a distance ≤ 110m guarantee the reception of at least 80% of the data packets transmitted through them.

These results let us conclude that, in order to use the disk model to define the set $\mathcal{L}$, the most appropriate value for the threshold is the one that provides the aforementioned percentage of packet reception. Henceforth we use the disk model with the specified threshold value.

B. Bandwidth capacity

The next parameter that we analyze is the bandwidth capacity of the network (Γ). This parameter sets a limit on the amount of data (measured as B/s) that can be successfully transmitted on the wireless channel.

In our model, the lower is the value of Γ, the less data can be transmitted by the nodes. As a result, more data may be dropped. However, a loose bound could produce routing policies that, when implemented, exceed the real bandwidth of the network. The excess of data increases the packet collisions and the wireless interference, hence producing more packet losses. Our goal is to precisely understand how the choice of Γ affects the network performance.

To this end, we consider different values of this parameter and examine their effects on the performance of the network. Again, we consider static scenarios because we focus on the stationary network performance. We also consider high data generation rates per node equal to $100KB/s$ in order to saturate the network and to make the effects of Γ more evident. We use a disk model to determine the sets $\mathcal{I}_i$. The distance threshold in this case is set to 200m which, under the given simulation settings, approximates the distance within two nodes can perceive each others' transmissions in the same wireless channel, i.e., they can interfere with each other. Further analysis regarding the choice of the sets $\mathcal{I}_i$ is subject for future work.

Discussion: Figure 2 (center) shows the simulation results. The data that are labeled "None" correspond to the instances where we did not make use of the bandwidth constraints. Since there were no significant differences regarding the delivery ratio of generated data, we consider as performance metric the delivery ratio in terms of the the data that were effectively transmitted. To this end we compute the delivery ratio of the transmitted data: the total number of packets received at the base stations divided by the number of packets generated and *actually sent* by the nodes (i.e., not dropped before transmission).

From the results, we can observe that conservative values of Γ induce a better delivery ratio of transmitted data because the channel is underutilized and, in general, free from collisions and interference. On the contrary, if we increase the Γ in the model we are implicitly allowing the network to transmit more data into the wireless channel, possibly exceeding the real bandwidth limits. As a consequence, much of this data will be lost and the delivery ratio decreases.

We conclude that a good estimation of the bandwidth of the network can be exploited by our model to provide solutions that make an efficient use of the wireless channel. Routing policies that respect the bandwidth limitations of the network do not increase the delivery ratio, but reduce the amount of data that are unnecessarily sent. This sort of flow control could help to save energy (e.g., due to less radio transmissions) and to avoid wasting resources.

C. Effect of link pruning in dynamic scenarios

To conclude the analysis, we study the effect of link pruning based on the time-to-live values. Our goal is to observe the effect of the threshold ttl^* that is used to prune the set $\mathcal{L}$ in dynamic scenarios.

To this end, we consider mobile scenarios with set of 10 mobile relays. We use different values for ttl^*. In addition, we also consider an alternative approach in which links are not removed from the model. Instead, each ttl_{ij} is used to derive a link weight γ_{ij} that favors the use of links with higher ttl values.

In the mobile scenarios, mission nodes travel through a set of predefined task locations. Upon the arrival at one task, they remain stationary for a random period of time, after which

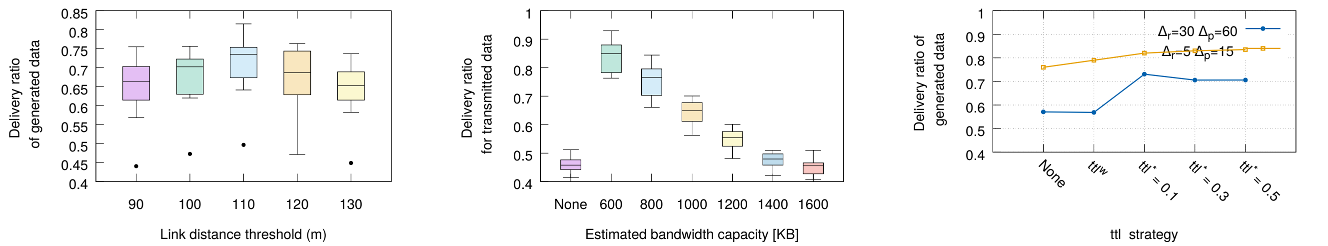

Fig. 2: Effect of the distance threshold (left), bandwidth capacity (center), and time-to-live threshold on the performance of the network. Results show the delivery ratio in terms of generated data (left and right), and transmitted data (center).

they select another task. Data generation increases when a robot is attending one task. With this simple setting we aim at recreating similar mobility and data rate scenarios such as the ones encountered in real-world robotic missions.

There is an interplay between the link pruning procedure and the frequency at which routing policies are computed. As mentioned in Section IV-C, the ttl values are defined with respect to the Δ_r. For long Δ_r time intervals, the fraction of links that have low ttl values may be higher compared to the case of small Δ_r (i.e., when routing policies are computed more frequently). To study this behavior, in our analysis we consider two different settings of Δ_p and Δ_r.

Discussion: Figure 2 (right) shows the results in terms of network delivery ratio under the two settings of Δ_p and Δ_r. The data that are labeled "None" correspond to the instances where we did not make use of the link pruning procedure. Data labeled ttl^w correspond to the approach where we do not use link pruning, but instead define the link weights based on the ttl values.

Results show that, when routes are updated frequently, the link pruning procedure does not significantly impact the performance of the network. This could be due to the small fraction of links with low ttl values. However, when routing policies are computed less frequently, the pruning of links plays an important role in the model. Moreover, we also note that the alternative approach of defining link weights does not provide any significant advantage.

These results let us conclude that using estimates of ttl of links can be successfully exploited in our model through the used link pruning procedure leading to better performance.

VI. Experimental results with a mobile robotic network

In this section we perform an experimental evaluation using a mobile robotic testbed with the goal of validating the framework. Before going into the the experiment results, we describe the setup and the implementation of the framework.

A. Experimental platform and software implementation

As experimental platform we use the foot-bot: a small differential drive robot (about 15 cm wide and 20 cm high). We equipped each foot-bot with two radio interfaces. An 802.11 based network operating on the 5 Ghz band allows to communicate directly with each robot and, for our purposes, plays the role of the low-bandwidth long range control network used for monitoring and control. A second wireless

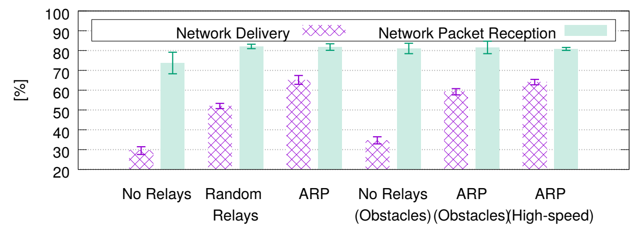

Fig. 3: Mean performance achieved for each one of the experimental scenarios considered. Values indicate the network delivery ratio and the network packet reception ratio as percentages. Error bars are one standard error of the mean.

interface is used as the data network. The wireless data network operates at the 2.4 Ghz band and its transmission power is artificially constrained in order to enforce the use of multi-hop routes in our lab environment. To this end, we use TL-WN722N Wi-Fi adapters with attenuators attached between the adapter and their external antennas.

To emulate a multi-hop wireless scenario we use a transmission power of 1 mW, a bit rate of 54 Mbps, and a signal attenuator of 20dBm. Empirically we determined that a link distance of 1.25m provides a link reception rate of at least 0.8 (see Section V-A). Therefore, we set this value to the link distance threshold for the preselection of wireless links.

The relay robots have a maximum speed of 0.3 m/s, while the mission robots move at a maximum speed of 0.1 m/s. A controller running on-board allows to navigate while avoiding collisions. We employ an OptiTrack motion capture system that provides the low level controllers with position information. The mobile robotic network testbed is deployed in an area of size 7 m $\times$ 7 m.

We consider 10 mission robots, 2 mobile relays, and 1 base station. We perform 6 scenarios. The first, only uses the mission robots. The second includes the relays, but these move in a random way. The third scenario makes use of our adaptive relay placement approach (ARP). The next two scenarios deal with cluttered environments. To this end, we placed 5 obstacles inside the area. These obstacles are considered inside the model as discussed in Section IV-C. Finally, we consider a more dynamic environment in which mission robots move with higher speed (0.2 m/s).

We performed 5 runs, lasting 10 minutes each, per scenario. Figure 3 shows the network performance. We observe that the network packet reception is uniform among the scenarios, and in overall high. This shows that the routing policies provide good performance. We recall that, in all the scenarios, the model is used to continuously update the routing policies. We can also observe the effect of including the relays into the system, and let the scheme to optimize their use. Deploying the relays in an adaptive way provided an increase of 15% with respect to the case when relays move randomly, and 30% with respect to the case where relays are not present. We also note that the performance level is maintained even in cluttered and more dynamic environments.

VII. Conclusions and Future Work

We have proposed a model-based approach for *dynamically controlling the position of the mobile relays* and computing at the same time *routing paths* for multi-hop data transmissions. The general goal is to *maximize the total throughput* of the data sent from the robots to the ground stations.

Our approach consists on using a MILP that explicitly takes into account and models effects of interference and congestion in wireless environments. The MILP exploits robots' positional information to estimate future network configurations and to derive estimates for the stability and reliability of the existing and prospective wireless links considered in the model. We demonstrate that the MILP is computationally affordable for reasonably sized scenarios. We studied the approach through realistic network simulations and fully validate the model with a mobile robotic testbed in various dynamic scenarios.

Future work includes the deployment of the framework in a team of humans, ground robots, and aerial robots in the context of outdoor search and rescue scenarios.

References

[1] D. Tardioli, D. Sicignano, L. Riazuelo, A. Romeo, J. L. Villarroel, and L. Montano, "Robot Teams for Intervention in Confined and Structured Environments," *Journal of Field Robotics*, vol. 7, no. PART 1, 2015.

[2] J. Fink, A. Ribeiro, and V. Kumar, "Robust Control of Mobility and Communications in Autonomous Robot Teams," *IEEE Access*, vol. 1, pp. 290–309, 2013.

[3] G. A. Hollinger and S. Singh, "Multirobot coordination with periodic connectivity: Theory and experiments," *IEEE Trans. on Robotics*, vol. 28, no. 4, pp. 967–973, aug 2012.

[4] A. Mosteo, L. Montano, and M. Lagoudakis, "Guaranteed-performance multi-robot routing under limited communication range," *Distributed Autonomous Robotics*, pp. 1–12, 2009.

[5] J. Thunberg and P. Ögren, "A Mixed Integer Linear Programming approach to pursuit evasion problems with optional connectivity constraints," *Autonomous Robots*, vol. 31, no. 4, pp. 333–343, aug 2011.

[6] O. Burdakov, P. Doherty, K. Holmberg, J. Kvarnstrom, and P.-M. Olsson, "Relay Positioning for Unmanned Aerial Vehicle Surveillance," *The Intl. Journal of Rob. Res.*, vol. 29, no. 8, pp. 1069–1087, apr 2010.

[7] E. I. Grøtli and T. A. Johansen, "Path Planning for UAVs Under Communication Constraints Using SPLAT! and MILP," *Journal of Intell. & Rob. Sys.*, vol. 65, no. 1-4, pp. 265–282, aug 2011.

[8] M. A. Hsieh, A. Cowley, V. Kumar, and C. J. Taylor, "Maintaining network connectivity and performance in robot teams," *Journal of Field Robotics*, vol. 25, no. 1-2, pp. 111–131, jan 2008.

[9] E. Feo Flushing, L. M. Gambardella, and G. A. Di Caro, "A mathematical programming approach to collaborative missions with heterogeneous teams," *Proc. of IEEE/RSJ IROS*, pp. 396–403, 2014.

[10] N. Bezzo, B. Griffin, P. Cruz, J. Donahue, R. Fierro, and J. Wood, "A Cooperative Heterogeneous Mobile Wireless Mechatronic System," *IEEE/ASME Trans. on Mech.*, vol. 19, no. 1, pp. 20–31, feb 2014.

[11] O. Cetin and I. Zagli, "Continuous Airborne Communication Relay Approach Using Unmanned Aerial Vehicles," *Journal of Intelligent & Robotic Systems*, vol. 65, no. 1-4, pp. 549–562, aug 2012.

[12] S. Gil, D. Feldman, and D. Rus, "Communication coverage for independently moving robots," in *IEEE/RSJ IROS*, vol. 1. IEEE, oct 2012, pp. 4865–4872.

[13] E. Stump, N. Michael, V. Kumar, and V. Isler, "Visibility-based deployment of robot formations for communication maintenance," in *Proc. of IEEE ICRA*. IEEE, may 2011, pp. 4498–4505.

[14] Y. Yan and Y. Mostofi, "Robotic Router Formation in Realistic Communication Environments," *IEEE Trans. Rob.*, vol. 28, no. 4, pp. 810–827, aug 2012.

[15] E. F. Flushing and G. a. Di Caro, "A flow-based optimization model for throughput-oriented relay node placement in wireless sensor networks," *Proc. of ACM SAC*, p. 632, 2013.

[16] D. Bhadauria, O. Tekdas, and V. Isler, "Robotic data mules for collecting data over sparse sensor fields," *Journal of Field Robotics*, vol. 28, no. 3, pp. 388–404, may 2011.

[17] P. Abichandani, H. Y. Benson, and M. Kam, "Decentralized multi-vehicle path coordination under communication constraints," in *Proc. of IEEE/RSJ IROS*. IEEE, sep 2011, pp. 2306–2313.

[18] J. Le Ny, A. Ribeiro, and G. J. Pappas, "Adaptive Communication-Constrained Deployment of Unmanned Vehicle Systems," *IEEE Journal on Sel. Areas in Comm.*, vol. 30, no. 5, pp. 923–934, jun 2012.

[19] S. S. Ponda, L. B. Johnson, A. N. Kopeikin, H.-L. Choi, and J. P. How, "Distributed Planning Strategies to Ensure Network Connectivity for Dynamic Heterogeneous Teams," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 5, pp. 861–869, jun 2012.

[20] L. Sabattini, N. Chopra, and C. Secchi, "Decentralized connectivity maintenance for cooperative control of mobile robotic systems," *The Intl. Journal of Rob. Res.*, vol. 32, no. 12, pp. 1411–1423, oct 2013.

[21] D. Tardioli, A. R. Mosteo, L. Riazuelo, J. L. Villarroel, and L. Montano, "Enforcing Network Connectivity in Robot Team Missions," *The Intl. Journal of Rob. Res.*, vol. 29, no. 4, pp. 460–480, apr 2010.

[22] R. K. Williams, A. Gasparri, and B. Krishnamachari, "Route swarm: Wireless network optimization through mobility," in *Proc. of IEEE/RSJ IROS*. IEEE, sep 2014, pp. 3775–3781.

[23] M. M. Zavlanos, M. B. Egerstedt, and G. J. Pappas, "Graph-theoretic connectivity control of mobile robot networks," *Proceedings of the IEEE*, vol. 99, no. 9, pp. 1525–1540, sep 2011.

[24] O. Tekdas, W. Yang, and V. Isler, "Robotic Routers : Algorithms and Implementation," *Int. Journal of Robotics Research*, pp. 1–23, 2010.

[25] M. Younis, I. F. Senturk, K. Akkaya, S. Lee, and F. Senel, "Topology management techniques for tolerating node failures in wireless sensor networks: A survey," *Computer Networks*, vol. 58, pp. 254–283, 2014.

[26] D.-T. Ho, E. I. Grøtli, P. B. Sujit, T. A. Johansen, and J. B. Sousa, "Optimization of Wireless Sensor Network and UAV Data Acquisition," *J. of Intell. & Robot Sys.*, vol. 78, no. 1, pp. 159–179, 2015.

[27] M. Kudelski, L. M. Gambardella, and G. A. Di Caro, "RoboNetSim: An integrated framework for multi-robot and network simulation," *Robotics and Autonomous Systems*, vol. 61, no. 5, pp. 483–496, 2013.

[28] C. Pinciroli, V. Trianni, R. O'Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. A. Di Caro, F. Ducatelle, T. Stirling, A. Gutierrez, L. Gambardella, and M. Dorigo, "ARGoS: a modular, multi-engine simulator for heterogeneous swarm robotics," in *Proc. of IEEE/RSJ IROS*, 2011, pp. 5027–5034.

[29] NS-3. Discrete-event network simulator for internet systems. [Online]. Available: http://www.nsam.org

[30] S. Gil, S. Kumar, D. Katabi, and D. Rus, "Adaptive communication in multi-robot systems using directionality of signal strength," *The Intl. Journal of Rob. Res.*, vol. 34, no. 7, pp. 946–968, jun 2015.