2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

A Mobility-Controlled Link Quality Learning Protocol for Multi-robot Coordination Tasks

Michal Kudelski, Luca M. Gambardella, Gianni A. Di Caro[1]

Abstract— **The performance of a team of robots executing a coordination task is, to a large extent, determined by the reliability of the communications between the robots. In wireless networks, one way to improve this reliability is to choose the best among the available wireless links. For this purpose, an accurate link quality model is required. We show how a group of robots can exploit their mobility to effectively and rapidly learn such a model directly from an unknown environment. The LQE (Link Quality Estimation) protocol, which is used by the robots to cooperatively collect labeled link quality samples, and learn out of them, is presented in the paper.**

The accuracy and the robustness of the LQE approach are validated through a set of real-world experiments, performed with mobile robots operating in different network environments. Moreover, in simulation, we study a multi-robot coordination problem, and show the benefits of using the link quality learning approach, at the expenses of devoting little time for learning the model before executing the task.

I. INTRODUCTION

The characteristics and the reliability of the network communication environment play a fundamental role shaping and affecting both behavior and performance of a multi-robot system [1]. This is especially true in the case of cooperative systems, which typically rely on information sharing to obtain coordinated behaviors and system-level synergies.

Considering the most general case in which the robots in the system are *mobile* and communicate through a *wireless multi-hop ad hoc network*, which can easily be very dynamic and quite unreliable [2], it becomes apparent that the robots need to operate some control on the communication environment to guarantee the effective and efficient functioning of the team. One general way to deal with this issue is to control robots' mobility to enforce them to preserve network connectivity (and quality) while they are executing some common task [3]. A complementary way of proceeding, which we propose in this work, consists in exploiting robots' mobility prior task execution, to gather network data about the general characteristics of the network environment the robot team will operate upon, and use this data to *learn* a regression mapping associating local network configurations to the quality of the wireless links present in the same configurations. In other words, we aim to define a *mobility-assisted protocol* to learn a *link quality estimator* [4] based on *supervised learning*: the observations gathered in a network environment where topology and traffic loads are controlled by the robots act as labeled samples which are used for learning a reliable link quality estimator. Hereafter we will refer to the protocol as *LQE* (Link Quality Estimator).

The availability of an accurate estimator can be used for multiple purposes in a robotic network. For instance, it can allow a routing protocol to robustly identify and select the best links to setup a good path to forward data from one robot to another; for the sake of maintaining network connectivity [3], [5] it can be used to estimate which area in the nearby space will provide a better communication quality-of-service and then let the robot move towards it. It is important to remark that, while the protocol can in principle be applied also to non-mobile networks, the use of controlled mobility makes LQE particularly suited for robotic networks, precisely because it can exploit their mobility.

The LQE approach presented in this paper is based on and extends the framework introduced in [6]. Link quality is evaluated in terms of the expected *packet reception ratio* (PRR), while *local network configurations* are defined through a set of *network features* determining the quality of a wireless link. The network features that we selected include: radio signal strengths, local topology, and local traffic characteristics of sender, receiver, and of their neighbors. LQE aims to learn the mapping between these set of features and links' PRRs.

By relying on a relatively small and easy to measure set of features, LQE does not directly address and use the modeling of complex channel phenomena such as multipath fading, shadow fading, signal interference, etc., which is an alternative and common way to proceed (as discussed in the related work section). However, since the effects of these phenomena have a direct impact onto the measured values of the selected network features, we claim that the effects get automatically and transparently incorporated into the learned regression mapping, without the need for the explicit use of mathematical models to capture their dynamics. These models are usually extremely complex (e.g., see [7]) and prone to be quite unreliable or of limited use in practice (especially in mobile scenarios), due to the many simplifying assumptions that are commonly required to build them. The mentioned phenomena depend in fact on the complex interplay between the characteristics of the used network technologies and those of the embedding physical environment (e.g., walls, obstacles, temperature, humidity). In Section VI we support our claim through a set of real-world experiments with mobile robots considering different environments, showing the excellent predictability and robustness achieved by LQE at the cost of a very low modeling complexity.

From an operational point of view, the LQE protocol, first *requires* the *definition* of a link quality metric and of the

[1]Authors are with Dalle Molle Institute for Artificial Intelligence (IDSIA), Lugano, Switzerland. E-mail: {michal,luca,gianni}@idsia.ch. G. A. Di Caro is corresponding author.

network features determining link quality (Sec. III). Then, the protocol consists of four steps:

1) *Collect* labeled link quality samples (Sec. IV);
2) *Learn* a link quality prediction model using a supervised approach (Sec. V);
3) *Deploy* the learned model to robots/sensors (Sec. VI);
4) *Use* the model for wireless communications (Sec. VII).

In the initial step, a group of mobile robots is deployed with the aim of collecting *labeled link quality samples* (i.e., pairs composed of a feature vector and the corresponding PRR value). Robots *move* in a controlled way, trying to maximize the number and the diversity of the observed local network topologies. At the same time, robots generate *probing messages* at variable rates, again with the aim of generating multiple network configurations but acting on the traffic side. By measuring the reception rate of the probing packets together with the values of the corresponding features describing the local network configuration, robots have all required information to make link quality samples.

After collected, the samples are used to learn a link quality model in the form of a regression mapping from the space of the network features to the PRR value. Once trained, the model is installed on each robot and can be used *online* to issue predictions about the expected quality of a link after assessing its related local network configuration. By exploiting its *generalization* capabilities, the regressor is able to estimate the expected PRR for a wide range of input configurations, including previously unobserved ones.

We validate the approach both on real networks (Section VI) and in simulation (Section VII). In the latter case, we consider a complex test scenario based on a *coordinated mobility* problem to show the effect and the importance of using the LQE learning approach in robotics. In the scenario, a few robots need to precisely coordinate their movements through message exchanging, but are out of direct communication range. Therefore, they need to use other robots to relay messages in a *multi-hop* routing modality. We show that even when a relatively short time period is used for collecting samples (a few minutes) and/or only a few robots are used, the use of the learned model is effective and significantly improves system performance.

II. Related Work

The work in this paper is partially based on the preliminary version of the protocol introduced in [6]. Here, we rely on the same framework, but we focus more on the mobility strategy and on the practical implementation of the protocol on robots, study more in depth the properties of the system, and show how the learned models can be directly employed to improve communication and coordination in cooperative multi-robot systems. Moreover, while in [6] we only report simulation results, here use mobile robots to validate LQE learning in multiple different real-world environments using a minimal set of easily measurable network features. A distributed version of the protocol was introduced in [8].

In the networking literature, the problem of wireless link quality estimation has been studied since long, and many approaches have been proposed so far. One comprehensive survey can be found in [4]. In general terms, the approaches used to characterize the elements affecting the performance of a wireless network, such as link quality, can be classified either as *analytical models* or *empirical estimators*.

Analytical models (such as [9], [10], [11]) try to capture wireless channel properties by building theoretical models, mostly based on the modeling of radio propagation and interference. For instance in [7], the authors address the spatial predictability of the wireless channel, which is a requirement for allowing a reliable learning approach as ours. They develop a complex probabilistic mathematical model taking into account multi-path and fading phenomena for predicting the spatial variations of the wireless channel, based on a relatively small number of measurements. While the model seems to produce good results, its application to dynamic radio environments such as those created by the presence of, and mutual interference among multiple robots is problematic and has not been shown or validated. In general, analytical models are typically very complex and are based on many assumptions. This makes extremely difficult to robustly apply analytical approaches in mobile and multi-agent scenarios, due to the complexity of interactions. Therefore, the validation is usually limited to simple static or quasi-static scenarios and its robustness is in general hard to assess. On the other hand, we propose a straightforward approach that automatically includes the complex channel properties in the learned regression model, and we show (through real-world experiments) that our method is robust and works well in different mobile multi-robot environments.

Empirical estimators can be categorized into two groups: *hardware-based* and *software-based* estimators. Hardware-based estimators obtain their measurements directly from the radio transceiver (such as the Received Signal Strength Indicator (RSSI), the Signal to Noise Ratio (SNR), and the Link Quality Indicator (LQI)). These measurements are easily available, yet it is known that they only allow to distinguish between very good and very bad links, and should not be used to classify intermediate quality links [4]. Software-based estimators are data-driven approaches, whose functionality is based on assessing the value of more complex measures, such as the packet reception rate (PRR) [12], [13], the required number of packet retransmissions (RNP) [14], [15], or any other quantity which is believed to have a correlation with the quality of a wireless link [16], [17], [18], [19]. We compared our approach to other empirical estimators in [8], where we demonstrated its advantages in terms of accuracy and faster adaptation capabilities.

In the robotic literature, the number of works addressing wireless links quality is relatively limited. Most of these works are related to connectivity issues. In [20], the authors propose to use SNR as a link quality measure in order to form optimal communication chains. In [21], the authors deal with the problem of maintaining connectivity in robotic sensor networks. In [22], mobile robots are employed to repair disconnected wireless sensor networks. Authors of [23] propose an interesting method for maintaining the link connectivity between a mobile robot and its control station.

To the best of our knowledge, none of the existing works neither directly exploits mobility nor try to learn a robust link quality model based on a set of basic, directly measurable network features (e.g., in [17] a supervised learning approach is used in sensor networks, but the used set of features are already preprocessed measures about congestion, queuing, and delivery probabilities). Furthermore, the benefits of link quality estimation on the performance of cooperative multi-robot systems have not been directly investigated so far, even if it is well understood that being aware of quality/bandwidth of a link is fundamental for an optimized allocation of network resources in a multi-robot system (e.g. [24]).

III. Network features for link quality

We consider the expected *packet reception ratio* (PRR) as a measure of link quality, defined as the ratio between received and sent data packets (i.e., PRR $\in [0,1]$). Nevertheless, the same approach can be applied, with appropriate adaptations, to any other notion of quality.

We assume that the PRR of a link is determined by its *local network configuration*. Based on the literature and our experience, we propose to represent this configuration by a vector of features that are relatively easy to measure and that correspond to main factors affecting the quality of wireless links. In particular, we consider the following features:

1) *distance* between the two end nodes of the link;
2) *Received Signal Strength Indicator (RSSI)* measured at the receiver node;
3) *Signal-to-Noise Ratio (SNR)* measured at the receiver;
4) *traffic load at the sender* r_S, and *traffic load at the receiver* r_R. These are the rates of the outgoing data traffic, monitored at each node. To ease learning, we assume that r_S and r_R belong to one of $n = 3$ possible *traffic profiles*: $T_{prof} = \{r_{low}, r_{med}, r_{high}\}$ (a different n could used depending on the application);
5) *neighborhood state of the receiver* V_R. V_R is a vector of elements $(n_i, RSSI_i,\ r_i)$, where n_i is an *active neighbor* (active in transmitting) of the receiver, transmitting with signal strength $RSSI_i$ and at a rate r_i (if RSSI is not available, distance or SNR can be used);
6) *neighborhood state of the sender* V_S (same as V_R).

In practice, the LQE protocol aims to learn a mapping between a vector of values of these network features and the expected PRR of the associated link. In other words, given a sender node S and a receiver node R, and given that they have around a number of other nodes active in transmissions, the learned LQE model will provide a robust prediction of the PRR that will be observed when packets will be sent from S to R in that local network configuration.

Feature preprocessing is performed to obtain a compact representation, appropriate for learning. For compacting V_R and V_S we proceed as follows. We define three RSSI profiles, as for the traffic profiles: $RSSI_{prof} = \{RSSI_{low}, RSSI_{med}, RSSI_{high}\}$ (distance or SNR profiles could be also used instead). A *feature set* is then defined as: $F = \{f_{ij} \mid i \in RSSI_{prof},\ j \in T_{prof}\}$. Using this definition the elements of the feature vectors V_R and V_S are conveniently grouped based on distance and traffic, and two compact feature sets, F_R and F_S, are defined to represent their data and encode the *neighborhood state*. For instance, the value of feature f_{ij} of set F_S is the number of elements of V_S that have RSSI profile $RSSI_i$ and traffic rate r_j. Assuming three different RSSI profiles and three traffic profiles as above, both F_R and F_S are composed of 9 numerical values. As a result, he final feature vector is:

$$\{distance, RSSI, SNR, r_S, r_R, F_S, F_R\}, \tag{1}$$

with 23 numerical values. Additionally, Each value is normalized in $[0,1]$, based on the empirically measured ranges.

IV. Collecting Samples with Mobile Robots

In LQE, the task of collecting labeled link quality samples (i.e., pairs of a feature vector and corresponding PRR value) is performed by the mobile robots as a group. The objective is to create as many network configurations as possible through *mobility*, and for each configuration the robots need to measure the values of all selected network features and the corresponding PRR value. The following *fully distributed sample collection protocol* implements this behavior.

At any time, a robot can be in one of the following three states: *Idle* (the robot is doing nothing), *Probing* (the robot periodically broadcasts probe packets and processes probe packets sent by other robots), and *Moving* (the robot performs controlled movements to enforce changes in the network topology). Transitions between states are performed on a random basis. The time between two consecutive state changes is a uniform random variable in $[min_s, max_s]$ (in the experiments, $min_s = 2$ sec, $max_s = 8$ sec). Every time a robot makes a state transition, it broadcasts a *state change* message to inform its neighbors about the new state.

A. Wireless channel probing at variable rates

When a robot switches to a *probing* state, it randomly chooses a traffic profile in $\{r_{low}, r_{med}, r_{high}\}$ (80Kbps, 400Kbps and 1.5Mbps, respectively). According to the profile, it periodically generates and locally broadcasts *probe* packets. Each probe packet contains: (i) a message counter, (ii) the rate at which the robot is transmitting packets, (iii) information about the robot's neighborhood, represented by the status vector V_S, (iv) the time stamp of the last change of the neighborhood state (the counter value at the moment of receiving a *state change* message from any of its neighbors).

In the *probing* state, robots also receive and process probe packets from their neighbors. When a probe packet from a given neighbor is received first, a *new sampling session* is started. Then, every received probe packet is examined and counted. A sampling session lasts until a change happens in the neighborhood (i.e., a session requires the stability of the local network environment). When a change in the local network configuration is detected (or the robot changes its own state), the affected sampling sessions are *closed*: sessions' PPRs are computed using the first and the last conuters successfully received, and the local network configuration is recorded as feature vector for the PPR. New sampling sessions can then start for the new network configuration.

B. Topology changes through robot mobility

By switching to *idle* and *moving* states, robots alter the network topology and change the local environment of the link. In idle state, a robot excludes itself from the network, reducing the number of nodes and creating a locally sparser network. Instead, robot movements modify the network structure by adding new links, and/or removing existing ones.

Robot mobility is exploited with the objective of continually reshaping network topology. Several mobility strategies can be used at this aim. To preserve distribution and simplicity of the protocol, we adopt a simple *random mobility* strategy to select robot trajectories: when a robot switches to the moving state, it randomly generates N possible destinations. Among them, it selects the one that is expected to generate the highest number of unobserved topologies, by exploiting the *memory of the network topologies* seen so far, as explained in the next section. Once selected the destination, the robot moves towards it following a possibly linear path with a constant speed of 10 m/s.

C. Using memory to maximize diversity of topologies

In order to maximize the diversity of observed network configurations, we let the robots use the following scheme to select their moves. We assume that we the information about robots' relative positions is accessible (e.g., from some GPS or other localization service). We define a *topology tuple* $T = \{distance_{level}, N_S, N_R\}$, where $distance_{level}$ is a discretized distance between sender and receiver (belonging to one of three distance ranges $\{dist_{low}, dist_{med}, dist_{high}\}$), N_S is a vector representing sender's neighborhood, and N_R is a vector representing receiver's neighborhood. Both N_S and N_R are composed of three values, corresponding to the numbers of neighbors within a given distance range. For instance, $N_S = \{1, 2, 3\}$ means that the sender has 6 neighbors: 1 within $dist_{low}$ (close range), 2 within $dist_{med}$, and 3 within $dist_{high}$. Hence, the tuple T is a simplified description of the local network configuration, taking into account only distances between the robots in the neighborhood.

Each time a new sampling session is opened, the corresponding topology tuple is stored in memory. When a robot selects a new destination, it derives the topology tuples that would be induced by moving towards the destination. By comparing these tuples with those in memory, it chooses the most exploratory destination. To make the process more efficient, robots share their memory with each other.

The *overhead* of the protocol is minimal: it requires one packet per second, of approximately 120 bytes, sent by each node. Training of the SVM requires less than 1 second on a standard PC for 300 samples (see [8] for related results).

A simple simulation experiment was performed to verify the efficiency of the proposed sample collection protocol. For 10 minutes, in an area of 450×450 m^2, samples have been gathered using various numbers of robots (10, 15, 20, and 25 robots). We are interested in the total number of collected

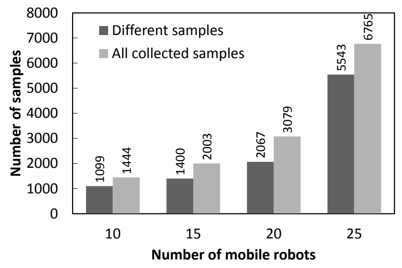

Fig. 1: Protocol efficiency: number and diversity of samples, collected for 10 minutes in an area of 450×450 m^2.

samples, but also in the diversity of the samples. In order to quantify the latter, we have discretized every collected sample to obtain a feature vector with only discrete values (i.e., we categorized the continues features — $distance$, $RSSI$, and SNR — into three classes each). Next, we have calculated the number of discretized samples differing in at least one feature. We refer to this number as the number of *different samples*. The results in Figure 1 shows that the robots are effectively able to collect a large number of samples within a limited time, and most of the samples (>80%) are relative to different network configurations.

V. Learning a Prediction Model

The training samples collected by robots are used to learn the regression mapping from the feature space to the PRR value. We assume that all the samples are sent to one processing unit which is responsible for training the model. It can be one of the robots, or an external machine able to communicate with robots and download the samples.

The machine learning model that we use is based on *Support Vector Machines* (SVMs) [25]. Namely, we use the ϵ-Support Vector Regression (ε-SVR) that non-linearly maps the input data features into a higher dimensional feature space using a kernel. ε-SVR is known to provide robust solutions towards regression problems characterized by high dimensionality, non-linearity, and local minima, which are the characteristics we expect for the network data we consider here. Moreover, compared to other possible approaches (e.g. neural networks), once learned, using an SVM to issue a prediction requires relatively little computations, making it suitable for using in real-time and embedded systems.

In order to check feature importance and eliminate possibly redundant and/or correlated features, we performed *feature ranking*. At this aim, we used the well-established feature selection approach of Weka [26], namely the *Information Gain* for evaluating the worthiness (weightage) of an attribute in the prediction task based on *entropy*. We observed that features $distance$, $RSSI$ and SNR are the most critical ones to differentiate good links from bad ones, but all features contribute with a significant weightage in predicting link quality. Thus, we decided to use all features described in Section III to train the regression models.

In additional experiments (not reported here) we observed that the use of either $distance$, $RSSI$, or SNR (in combination with the features in Section III) is needed to achieve high prediction accuracy, but they do not need to be considered altogether to obtain a good performance. This can be understood by observing that the three features are all a function of each other. Therefore, a convenient selection among $distance$, $RSSI$ and SNR can be made, if required to make the learning model simpler/lighter and/or if one of the features (e.g., distance) cannot be measured in practice.

In order to use ε-SVR, some of the internal parameters need to be tuned. In our work we employ the widely used *Radial Basis Function* (RBF–Gaussian) as a kernel. Thus, three essential SVR parameters need to be adjusted to data: an error penalty parameter C, a Gaussian RBF kernel parameter γ, and a loss function parameter ε. A standard k-fold cross-validation procedure was used to set the values of these parameters based on a data set of 200’000 samples, resulting in $C = 32, \gamma = 0.25, \varepsilon = 0.0625$.

VI. Real-world Validation of LQE Learning

The proposed link quality estimation approach makes sense only when the considered features are both easily accessible and capture most of the channel characteristics that affect the effective quality of a wireless link. These characteristics include complex and difficult to model phenomena, such as shadow fading, multipath fading, signal interference, etc. In this section, we validate our link quality learning approach through a set of real-world experiments.

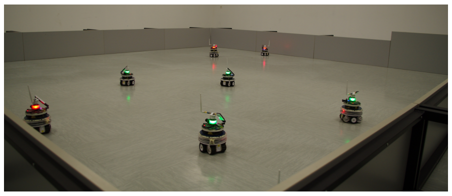
Fig. 2: Open space experiment: modified *foot-bots* at work.

We employ a set of *foot-bots*, small ground robots developed within the project Swarmanoid (`www.swarmanoid.org`), and have equipped with TL-WN722N Wi-Fi adapters with external antennas. To reduce the transmission range, we use signal attenuators (2x20 dB) attached between wireless adapters and their antennas (see Fig. 2). To set up the experiments minimizing external noise influence, we used the *Wi-Spy 2.4x* spectrum analyzer as a monitor. Robots move at the speed of 0.3 m/s and follow a *random waypoint mobility model* (pause time: 15–25 seconds). Network traffic is generated according to three traffic profiles: $\{r_{low}, r_{med}, r_{high}\}$ (30Kbps, 80Kbps and 160Kbps, respectively).

We considered three different communication environments, aiming to show that in each case we are able to learn the mapping from the space of features to the PRR value. If the learned mappings provide accurate predictions of PRR, then the considered features provide enough information on channel characteristics (with respect to the task):

1) *Open Space environment.* It aims to represent a large open-space area with no obstacles. For this purpose, we limit the transmission range to 2 meters, and we place 12 robots in a bounded area (8m$\times$6m) within a bigger room (Fig. 2). Nearest obstacles are separated by more than the transmission range, thus we effectively may consider the environment as an open space;
2) *Noisy indoor environment (Robo Lab).* 8 robots are placed in bounded area of 4m$\times$3m (tx-range: 1.5m) within a large room with multiple obstacles around (walls, metal racks, etc.) and multiple sources of noise;
3) *Multi-floor, multi-room indoor environment (INDRIYA).* We use the *INDRIYA Testbed* [27] composed of 139 static wireless sensor nodes. We simulate mobility by considering 40 logical nodes that change their position every 3 minutes (see [8] for details).

We consider the following network features, which are immediately accessible in almost any networked system (i.e., we are not considering $distance$, which might not be always accessible): $\{RSSI, r_S, r_R, F_S, F_R\}$ (see Sec. III). In each experiment, we collect a set of samples with their corresponding timestamps. In order to observe the evolution of the learning process, we retrain the SVR model each time we collect 10 new samples (we always use all the samples collected so far). We calculate the prediction error for each sample, before it is added to the training set. Based on these prediction errors, we measure the evolution of the *mean squared error (MSE)* as a moving average of the last 50 samples. We also report the final MSE value and the boxplot of the absolute prediction errors, both calculated after the initial learning phase (after the MSE value stabilizes over time). The results are shown in Figure 3.

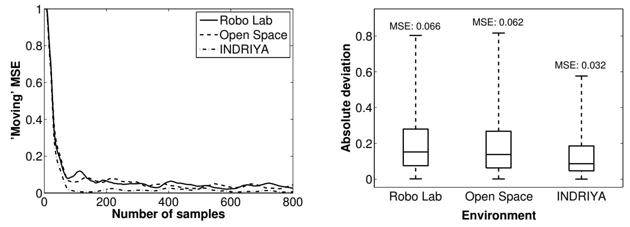

Fig. 3: Evolution of prediction accuracy: window averaged MSE (left). Final prediction accuracy: distribution of absolute prediction errors calculated after 300 samples (right).

We can observe that after collecting 300 samples the measured prediction accuracy is quite good in all the considered different environments. Collecting 300 samples took 20 minutes in the INDRIYA testbed, and 30 minutes in the other two testbeds. From Figure 3 (left), it is also clear that already after 100 samples (for INDRIYA) and 150 samples (for RoboLab/OpenSpace) the prediction accuracy is more than satisfactory. In practice, 15 minutes seem to be the average time needed to collect enough samples to learn an acceptable mapping in all the different environments. The INDRIYA testbed appeared to be the ’easiest’ environment. This can be explained by the fact that we actually considered a sequence of static topologies there. In turn, this also shows that disturbances introduced by mobility (e.g., due to vibrations, rotating antennas, etc.) is significant.

In conclusion, the experimental results support the claim that our learning approach allows to build a robust mapping from simple local measurements to the expected link quality in complex communication environments, even though we do not directly analyze the impact of complex phenomena such as multipath fading or signal interference. These phenomena are *indirectly* incorporated into the features and the learned regression mapping. Moreover, as we both learn and predict based on measured values (e.g., the RSSI), our methodology is supposed to be robust to local environment changes. These will mostly affect signal propagation, which will be indirectly incorporated in our measurements, and thus considered by the learned model. This was partially demonstrated in the mobile experiments, in which robots themselves are dynamic obstacles that generate noise and reflections for others.

VII. Test Case Scenario: Coordinated Mobility

Once trained, the link quality model can be installed in every robot, providing access to link quality estimates for various applications and network protocols. The only requirement is to provide as input the current values of the features describing the local environment. This can be easily achieved by performing local measurements and exchanging basic information about the local network configurations between the robots (e.g., through a low-rate broadcasting).

As a test case scenario to demonstrate the benefits of using the LQE approach, we consider a multi-robot system and a relatively simple *coordination problem*. We define a challenging communication environment and we design the task aiming to emphasize the impact of one-to-one unicast communication between two specific robots set spatially apart in the network. Two communication approaches for *data routing* are compared under various performance metrics: with and without link quality learning.

In both cases, the *OLSR* [28] *routing protocol* for mobile ad hoc networks is used as a reference algorithm. OLSR is one of the main Internet standards for mobile ad-hoc and mesh networks. It is a *proactive link-state* routing protocol, which means that each node periodically broadcasts the information about its links to its neighbors. This information is propagated through the network, and is used by other nodes in to build their view of the *connection graph* for the whole network. The graph is in turn used to calculate routing tables. In its basic version, OLSR simply minimizes the number of hops between source-destination pairs (i.e., it assumes that the cost of each connection in the graph is equal to 1). Here, we show that replacing this simple routing metric with a metric based on the estimated quality of wireless links can significantly improve the communication performance. More precisely, we propose to use the metric based on the *Expected Transmissions Count* (ETX): a cost of a single wireless link is defined as the expected number of transmissions required to send a single packet through this link (ETX can be calculated as the inverse of PRR, see [29] for details). The cost of a path composed of many wireless links is the sum of the ETX of the individual links. In what follows, we refer to our implementation as to *OLSR-LQE*.

In order to ensure realistic system-level simulations, we use the integrated simulation environment *RoboNetSim* proposed in [1], which combines a multi-robot simulator (ARGoS [30]) with a realistic network simulator (NS-3 [31]).

A. Problem Description

Two selected robots need to coordinate their actions remotely. One of them is termed *master* and defines the mobility pattern. The other is termed *slave* and is supposed to imitate the same mobility pattern of the master, but on a different area. For simplicity, we assume that robots are aware of their own position and orientation. The master periodically sends the information about its current destination to the slave (once per second). The master and the slave are out of their communication range. Hence, they need to use the other robots in the system to relay their messages. The other robots perform their own tasks and also communicate with each other. This is simulated by letting them move according to random mobility patterns (based on the Random Waypoint mobility model) and generating data according to random traffic generation patterns (a subset of the robots generate CBR traffic to random destination robots).

As a result, a mobile ad-hoc network is formed among the robots. To make the task even more difficult, we allow for K *master-slave pairs*. The performance of the system is evaluated in terms of the absolute error between the desired and the actual position of a slave (averaged over K slaves), which is strongly affected by the ability of the master-slave pairs to effectively communicate in in multi-robot mobile network. We consider two communication strategies, to show the effect of using LQE:

- *OLSR:* all robots are immediately deployed for task execution. They use the standard OLSR routing daemon, without link quality extensions;
- *OLSR-LQE:* before robots are deployed to execute the task, a subset of robots is used to collect link quality samples and to train a link quality model. Then the model is installed in all robots, and they start executing the task using OLSR based on link quality estimates.

B. Experimental Settings

1) Simulation environment: In NS-3, we simulate 802.11a Wi-Fi networks, with the transmission rate of 6 Mbps. We use a log distance propagation loss model with default parameters (path loss exponent set to 3.0). This corresponds to a transmission range of roughly 120 m. The UDP protocol is used in the transport layer. In ARGoS, we set the simulation step to 0.1s and we use a 2D dynamics physics engine.

2) Scenario: We simulated the above scenario for 34 foot-bots (30 robots + 2 masters + 2 slaves). Robots move within a wall enclosed area of 500×500 m^2 , with a maximum speed of 1 m/s (only masters and slaves move faster: 10 m/s). 15 nodes are used as *constant bit rate* (CBR) sources, each one generating 50 packets per second (packet size is set to 70 bytes). The simulation is executed for 1000 seconds. Results are averaged over 20 simulation runs. We analyzed the averaged absolute error between the desired and the actual position of a slave. We observed this error as a function of time, as well as we studied the distribution of the absolute error, visualized as the empirical cumulative distribution function (CDF). The distance between the master and the slave is set to 375 meters. Therefore, a minimum number of 4 hops is required for communication.

3) Link quality learning: The parameters of the sample collection protocol are summarized in Table I. For learning we used 25 mobile nodes deployed in an area of 600×600 m^2, gathering 200,000 samples. The *mean squared error* obtained while training was equal to $MSE = 0.02$.

TABLE I: Parameters of the sample collection protocol.

min_s	2 secs
max_s	8 secs
$\{r_{low}, r_{med}, r_{high}\}$	$\{80, 400, 1500\}$ [Kbps]
$Speed$	10 [m/s]
N	10

C. Impact on coordination

In the first experiment, we examined the impact of the link quality learning approach on the performance coordination. For OLSR-LQE, we used a reference link quality model trained according to the settings from the previous section.

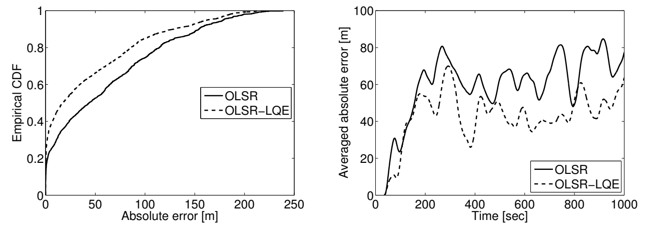

Fig. 4: Performance in terms of the absolute error between the desired and the actual position of the slave. The error as a function of time (right), and the error distribution as an empirical CDF (left). Results for 34 robots, link quality model trained on 200'000 samples (using 25 mobile robots).

From the results in Figure 4, it can be observed that the task is quite difficult. The average position error vary within the range of 40 to 70 meters. This is implied by the demanding communication environment: multi-hop communication via mobile relaying nodes resulted in an average packet delivery ratio of only 50%. Thus, it could happen that slaves did not receive any information from their masters for a certain amount of time, due to lack of available routing paths. In such an environment, the way a routing mechanism selects wireless links to transmit data can have a remarkable impact on communication performance and, consequently, on the resulting performance of the multi-robot system.

This is precisely demonstrated when comparing the results of OLSR and OLSR-LQE. When we measured the actual packet delivery ratio between slave and master robots, OLSR-LQE obtained on average 30–40% better results. Accordingly, the coordinated mobility task is solved evidently better by the robots that benefit from the link quality model.

D. Effect of time/robots resources

In the next two experiments, we investigate how much effort is actually required in order to really benefit from the link quality learning approach. In other words, we want to analyze the efficiency of the proposed mobility controlled sample collection mechanism by answering the following question: how much time and how many mobile robots do we need to collect the amount of information which is required to improve communications?

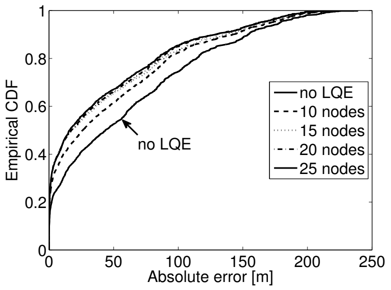

Fig. 5: Performance comparison for using 10, 15, 20, and 25 robots to collect samples for 10 minutes (collected 1444, 2003, 3079, and 6765 samples, respectively).

In the first experiment (Fig. 5), sample collection time is limited to 10 minutes and the effect of using different number of robots for collecting samples (10, 15, 20 and 25 robots, deployed in areas of 350×350, 450×450, 500×500 and 600×600 m^2, respectively) is considered. Despite the reduced number of collected samples, it results that is still possible to learn good models. For 15, 20, and 25 robots, the learning performance is practically the same, showing no deterioration (at least from the perspective of the task). For 10 collecting robots, the performance is slightly worse, yet still better of that obtained with OLSR.

These results show that the number and the diversity of the collected samples, combined with the generalization capabilities of the ε-SVR learning model, were sufficient to learn a good regression mapping. It is a remarkable result, taking into consideration the following facts. We measured that during 1000 seconds of task execution, on average a total number of roughly 90'000 different local network configurations is observed (different in terms of the corresponding discretized feature vectors, as discussed in Section IV). On the other hand, the number of configurations observed during training is much lower (see Figure 1). This means that: (i) the samples collected for training are indeed well diversified and representative, and (ii) the learning model and the used parameter values provide a truly good generalization.

In the second experiment (Fig. 6), we used 15 robots and we collected samples for a limited time of 3, 5, and 10 minutes. The results are in line with those of the previous experiment, confirming the generalization capabilities of our learning approach. They also show that it is beneficial to spend even a small amount of time (such as 3 or 5 minutes) on learning the link quality model. Moreover, it can be observed that in the considered environment it is enough to use 15 robots and to collect samples for 10 minutes: more extensive sample collection does not offer noticeable profit.

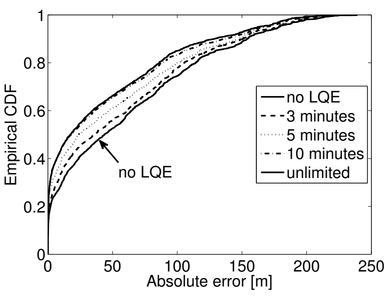

Fig. 6: Performance comparison for different collection times using 15 robots: 3, 5, 10 minutes, and unlimited (collected 773, 1186, 2003, and 200'000 samples, respectively).

VIII. Conclusions and Future Work

We have presented the LQE (Link Quality Estimation) protocol, which aims to learn a reliable quality model for wireless links in multi-robot systems. By exploiting robots' mobility and data traffic generation in a controlled way, LQE can efficiently generate many different network configurations. For each configuration, the robots can perform sets of measurements related to local network conditions, such as traffic loads, RSSI, and SNR. These measurements are used as input to a supervised machine learning approach that allows to learn a regression mapping from local network configurations to expected link quality.

Through real-world experiments using mobile robots, we have shown that the use of LQE allows to reliably learn accurate link quality prediction models. Moreover, the model was employed to improve the communication within a group of mobile robots performing a coordination task: even when limited time or robot resources were engaged, the use of LQE could significantly improve system performance.

As future work, we aim to use the learning model to make spatial predictions of link quality, and use these predictions to locally optimize the trajectories of mobile robots (e.g., maximize the QoS of the communications along the path). We will also perform a more in depth sensitivity analysis to better assess the impact of the different features.

References

[1] 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.

[2] C. Tschudin, P. Gunningberg, H. Lundgren, and E. Nordström, "Lessons from experimental manet research," *Elsevier Ad Hoc Networks Journal*, vol. 3, no. 2, pp. 221–233, 2005.

[3] D. Tardioli, A. Mosteo, L. Riazuelo, J. Villarroel, and L. Montano, "Enforcing network connectivity in robot team missions," *Int. Journal of Robotics Research*, vol. 29, no. 4, pp. 460–480, 2010.

[4] N. Baccour, A. Koubâa, M. Zuniga, H. Youssef, C. Boano, and M. Alves, "Radio link quality estimation in wireless sensor networks: a survey," *ACM Transactions on Sensor Networks*, vol. 8, no. 4, 2012.

[5] Y. Mostofi, M. Malmirchegini, and A. Ghaffarkhah, "Estimation of communication signal strength in robotic networks," in *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 1946–1951, 2010.

[6] E. F. Flushing, J. Nagi, and G. A. Di Caro, "A mobility-assisted protocol for supervised learning of link quality estimates in wireless networks," in *Proceedings of the Int. Conf. on Computing, Networking and Communications (ICNC), Workshop on Mobility and Communication for Cooperation and Coordination (MC ^{3})*, 2012.

[7] M. Malmirchegini and Y. Mostofi, "On the spatial predictability of communication channels," *IEEE Transactions on Wireless Communications*, vol. 11, no. 3, pp. 964–978, 2012.

[8] G. A. Di Caro, M. Kudelski, E. Feo, J. Nagi, I. Ahmed, and L. Gambardella, "On-line supervised learning of link quality estimates in wireless networks," in *Proc. of 12th IEEE/IFIP Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, pp. 69–76, 2013.

[9] P. Gupta and P. Kumar, "The capacity of wireless networks," *IEEE Trans. on Information Theory*, vol. 46, no. 2, pp. 388–404, 2000.

[10] J. Padhye, S. Agarwal, V. N. Padmanabhan, L. Qiu, A. Rao, and B. Zill, "Estimation of link interference in static multi-hop wireless networks," in *Proc. of ACM SIGCOMM IMC*, p. 1, 2005.

[11] A. Iyer, C. Rosenberg, and A. Karnik, "What is the right model for wireless channel interference?," *IEEE Trans. on Wir. Comm.*, vol. 8, pp. 2662–2671, May 2009.

[12] A. Woo and D. Culler, "Evaluation of efficient link reliability estimators for low-power wireless networks," tech. rep., EECS Department, University of California, Berkeley, 2003.

[13] M. Senel, K. Chintalapudi, D. L., A. Keshavarzian, and E. Coyle, "A kalman filter based link quality estimation scheme for wireless sensor networks," in *Proc. of IEEE GLOBECOM*, pp. 875–880, 2007.

[14] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *Proc. of MobiCom '03*, pp. 134–146, ACM Press, 2003.

[15] R. Fonseca, O. Gnawali, K. Jamieson, and P. Levis, "Four-bit wireless link estimation," in *6th Workshop on Hot Topics in Networks*, 2007.

[16] K. Farkas, T. Hossmann, F. Legendre, B. Plattner, and S. Das, "Link quality prediction in mesh networks," *Computer Communications*, vol. 31, pp. 1497–1512, may 2008.

[17] Y. Wang, M. Martonosi, and L.-S. Peh, "Predicting link quality using supervised learning in wireless sensor networks," *ACM SIGMOBILE Mobile Computing and Comms. Review*, vol. 11, no. 3, p. 71, 2007.

[18] N. Baccour, A. Koubâa, H. Youssef, M. Ben Jamâa, D. do Rosário, M. Alves, and L. Becker, "F-LQE: A fuzzy link quality estimator for wireless sensor networks," in *Proc. of 7th European Conference on Wireless Sensor Networks*, pp. 240–255, 2010.

[19] T. Liu and A. E. Cerpa, "Foresee (4C): Wireless link prediction using link features," in *Proc. of IPSN'11*, pp. 294–305, 2011.

[20] C. Dixon and E. W. Frew, "Controlling the mobility of network nodes using decentralized extremum seeking," in *Proc. of the 45th IEEE Conference on Decision and Control*, pp. 11–12, 2006.

[21] W. Zhuang, X. Chen, J. Tan, and A. Song, "An empirical analysis for evaluating the link quality of robotic sensor networks," in *Proc. of Int. Conf. on Wireless Comm. Signal Processing (WCSP)*, pp. 1–5, 2009.

[22] K. Luthy, E. Grant, and T. Henderson, "Leveraging rssi for robotic repair of disconnected wireless sensor networks," in *Proc. of IEEE Int. Conf.e on Robotics and Automation (ICRA)*, pp. 3659–3664, 2007.

[23] N. Pezeshkian, J. D. Neff, and A. Hart, "Link quality estimator for a mobile robot," in *Proc. of 9th Int. Conference on Informatics in Control, Automation and Robotic (ICINCO)*, pp. 87–94, 2012.

[24] C. Mansour, E. Shammas, I. Elhajj, and D. Asmar, "Dynamic bandwidth management for teleoperation of collaborative robots," in *Proceedings of the IEEE Int. Conf. on Robotics and Biomimetics*, 2012.

[25] V. N. Vapnik, *The nature of statistical learning theory.* New York, NY, USA: Springer-Verlag New York, Inc., 1995.

[26] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten, "The WEKA data mining software: an update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[27] M. Doddavenkatappa, M. C. Chan, and A. L. Ananda, "Indriya: A low-cost, 3D wireless sensor network testbed," in *Proceedings of TRIDENTCOM*, pp. 302–316, 2011.

[28] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized link state routing protocol for ad hoc networks," in *Proc. of the IEEE Int. Multi Topic Conf. Technology for the 21st Century (INMIC)*, pp. 62–68, 2001.

[29] OLSRd, "An adhoc wireless mesh routing daemon." `http://www.olsr.org`.

[30] 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 the 24th IEEE/RSJ IROS*, pp. 5027–5034, 2011.

[31] NS-3, "Discrete-event network simulator for Internet systems." `http://www.nsnam.org`.