

# Measuring the Effects of Communication Quality on Multi-robot Team Performance

Tsvetan Zhivkov<sup>1</sup>(✉), Eric Schneider<sup>1,2</sup>, and Elizabeth I. Sklar<sup>1</sup>

<sup>1</sup> Department of Informatics, King's College London, London, UK  
{[tsvetan.zhivkov](mailto:tsvetan.zhivkov@kcl.ac.uk),[eric.schneider](mailto:eric.schneider@kcl.ac.uk),[elizabeth.sklar](mailto:elizabeth.sklar@kcl.ac.uk)}@kcl.ac.uk

<sup>2</sup> Department of Computer Science, University of Liverpool, Liverpool, UK

**Abstract.** Maintaining network connectivity is crucial for multi-robot and human-robot teams. If robots lose their network connection, they cannot receive commands or share sensor data with teammates. Most research in the multi-robot systems and human-robot interaction communities assumes 100% network connectivity, 100% of the time; but this is unrealistic for real-world domains. Indeed, this assumption could be associated with significant risk, depending on the robots' task domain. This paper presents preliminary results for measuring the impact of communication loss on multi-robot team performance. A series of controlled experiments were conducted, with physical and simulated robots, where the probability of packet loss is gradually increased from 0% to 75%. The experiments show that the multi-robot team exhibits a non-linear decrease in performance with respect to an increase in percentage of packets dropped.

**Keywords:** Multi-robot team · Communication · Empirical results

## 1 Introduction

Consideration of *communication network quality* is a real and significant issue in multi-robot and human-robot teams. An unreliable network connection can mean that messages get dropped and robots lose their ability to receive commands, transmit sensor data and generally interact with either human or robot teammates. In real-world settings, mobile phones drop calls and bandwidth degrades when signal strength declines, even with high-speed 4G networks. As autonomous mobile robots transition from research laboratories into operational environments such as factories, hospitals, schools and homes, it becomes imperative that they are able to communicate reliably with those around them, whether human or robot. Humans in close proximity rely on non-digital forms of communication, such as speech and gestures; and there is much attention paid, in human-robot interaction and artificial intelligence, to the investigation of methods for robots to communicate in similar ways, using speech recognition, natural language generation and gesturing. In addition, robots will frequently be deployed in task domains where they are not co-located with humans, such as search-and-rescue, humanitarian de-mining or nuclear plant monitoring. It is these types of non-proximal relationships that are of concern in our work.

The study presented here examines the impact of degrading communication quality in a multi-robot team. The long term goal of this line of work is to quantify the effects and assess the risks associated with poor communication quality on multi-robot and human-robot teams in a physical environment. In the study described in this paper, we conducted a series of experiments in which the success of message transmission in a multi-robot team is not guaranteed. Our working hypothesis is that poor-quality communication will hinder the execution of tasks which require interaction amongst robots or between robots and local server(s), for example, where mission-critical information is stored, such as a detailed map of the robot's environment. To evaluate this hypothesis, a simple method of random packet loss was implemented and applied to infect communication functions, to measure how a robot team copes with poor communication. Our overarching objective is to demonstrate how changes in communication quality impact task performance, using a number of metrics and experimental conditions. Experiments were conducted in simulation and with physical robots. The experiments show that the multi-robot team exhibits a decrease in performance with respect to an increase in percentage of packets dropped. This result is not unexpected, but the contribution of this work is that now we have a framework and results that allow us to quantify the impact of communication loss on specific performance metrics.

The remainder of this paper is organised as follows. Section 2 presents our approach and describes details of our methodology. Section 3 outlines our experimental setup, including the software and hardware environment developed for empirical studies. Section 4 presents the results and statistical analysis of our experiments. Section 5 briefly highlights related work in multi-robot and human-robot systems. Finally, Sect. 6 summarises our results and mentions our immediate next steps and future directions with this line of research.

## 2 Methodology

In order to experiment with the notion of packet loss in a multi-robot team, we implemented a probabilistic function that is interjected into our system's messaging server (detailed in the next section). This function will fail to send messages, at a rate no greater than a value passed to the function. In other words, in order to simulate 25% packet loss, this function is called with a value of 0.25; and inside the function, messages are only transmitted 75% of the time. During an experimental run, the chosen packet-loss value remains static until the end of that experiment.

An experimental schema  $F$  is defined as a tuple:  $F = \langle N, T, S, P \rangle$  where  $N$  is the size of the robot team (number of robots);  $S$  is a specific experimental *scenario*;  $T$  is a set of tasks; and  $P$  is a simulated loss in communication quality. The values used for the experiments described here are:

- $N = 2$
- $T = 6$

- $S = \{S_1, S_2\}$
- $P = \{0\%, 25\%, 50\%, 75\%\}$

Each scenario,  $S$ , is defined as having a map  $M$ , a team of robots with starting locations  $\{(x_1, y_1) \dots (x_N, y_N)\}$  and a set of task locations  $\{(x_1, y_1) \dots (x_T, y_T)\}$ . Following other related work [4, 7, 12, 13], the tasks and locations can be classified as: single-robot (SR) vs multi-robot (MR) tasks; clustered start (CS) vs distributed start (DS) locations; independent (IT) vs constrained (CT) tasks; static (SA) vs dynamic (DA) arrival of tasks; instantaneous (ID) vs extended (ED) task duration. In the work presented here, we restrict the scenarios to  $\langle SR, CS, IT, SA, ID \rangle$ ; although future work will explore the full range of types of tasks.

### 3 Experiments

Our experiments were conducted using the *MRTeAm* framework [12], which integrates ROS [9] controllers for individual robot navigation with the RabbitMQ<sup>1</sup> messaging system for handling inter-robot communication. The ROS navigation stack provides communication, localisation and path-planning capabilities. A key feature of ROS is that the same robot controllers can be used in a simulated environment as well as with physical platforms. We used the Stage [3] environment for simulation experiments, which includes an emulator for our physical platform: the Turtlebot2<sup>2</sup>. This robot has a differential drive base and an RGB-depth camera (the Asus Xtion<sup>3</sup>, which is a clone of the Microsoft Kinect [10]), and an on-board laptop (the Acer Travelmate B117<sup>4</sup> running Ubuntu). The robots communicate with each other by sending and receiving messages via WiFi. Our RabbitMQ service runs on a local server.

The operating environment for our robots is an office setting with rooms opening off of a circular corridor (illustrated in Figs. 4 and 5). The corridor has multiple sets of fire doors, which typically stay closed. Thus our physical experiments were restricted to that portion of the corridor which could be reached without needing to open the fire doors. We use the campus WiFi for our robots, which is a local instantiation of *eduroam*<sup>5</sup> that supports IEEE 802.11b and 802.11g standards.

For our experiments, we had our robot team execute missions consisting of exploration tasks in which the robots went to particular locations, ostensibly to perform surveillance tasks, though the experiments described here only involved the robots travelling to their assigned locations. At the beginning of a mission, robots received from the server a set of tasks—locations to visit. While robots moved to each of their assigned task location(s), messages<sup>6</sup> passed between the

<sup>1</sup> <https://www.rabbitmq.com/>.

<sup>2</sup> <http://www.turtlebot.com/>.

<sup>3</sup> [https://www.asus.com/3D-Sensor/Xtion\\_PRO\\_LIVE/](https://www.asus.com/3D-Sensor/Xtion_PRO_LIVE/).

<sup>4</sup> <https://www.acer.com/ac/en/GB/content/professional-series/travelmateb>.

<sup>5</sup> <https://www.eduroam.org/>.

<sup>6</sup> Specifically, the messages that were passed were ROS `amcl_pose_msg` messages.

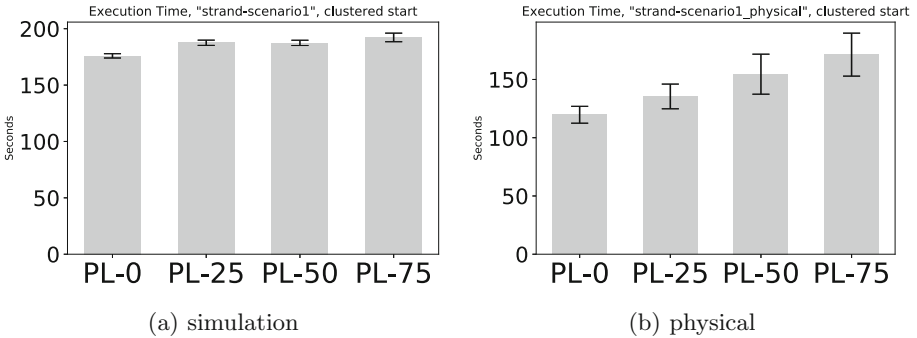
robots, via our server (running RabbitMQ), facilitating the robots' sharing their locations ("pose" information). This helps the team members know where each other are in the environment during a mission.

During an experiment, data are obtained by using a particular schema  $F_i$  with a given scenario and packet-loss value. In total eight different schema configurations are identified ( $N \times T \times S \times P$  with  $|N| = 1, |T| = 1, |S| = 2, |P| = 4$ ). Two scenarios,  $S_1$  and  $S_2$  (illustrated in Figs. 4 and 5) were defined for running experiments in simulation and with physical robots, respectively. The packet loss parameter  $P$  has four allocated values,  $\{0\%, 25\%, 50\%, 75\%\}$  (also denoted as PL-0, PL-25, PL-50 and PL-75, respectively). For each configuration, a total of 25 simulations and 20 physical runs were obtained.

### 3.1 Performance Metrics Rationale

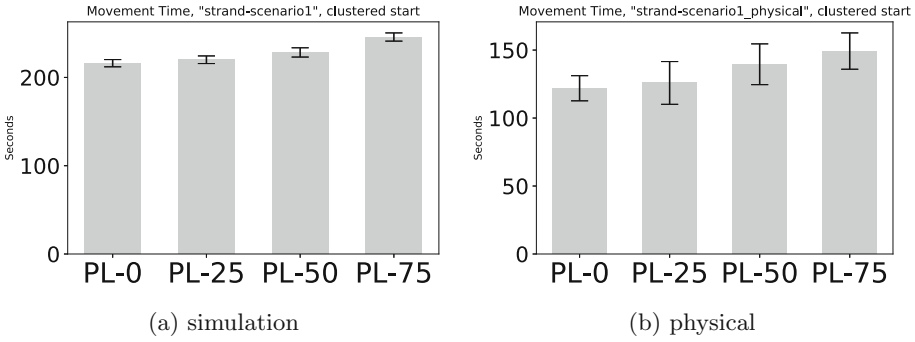
The MRTeAm framework collects thirty-two performance metrics in all, recorded after each experimental run. Only the parameters relevant to the research question considered here are reported in this paper, namely: **execution time**, **total movement time**, **total distance travelled**, **near collisions**, **delay time** and **idle time**. These are described, in turn, below.

The **execution time** is the total time taken for the robot team to complete all the tasks allocated to them. As can be seen in Fig. 1a, for the simulation runs, the execution time increases with each increase in parameter  $P$ . For the physical runs, shown in Fig. 1b, the trend is similar but with greater variance in results. The increase in variance is as expected, and it is a result of the noisy physical environment.



**Fig. 1.** Results for *execution time*, with increasing  $P$

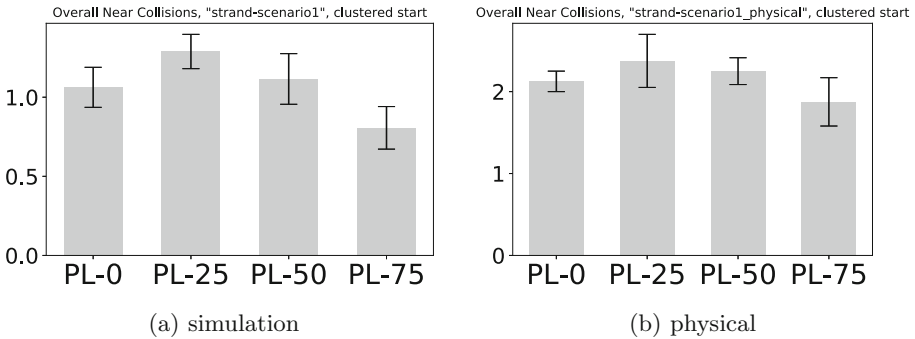
The **total movement time** metric (Fig. 2) shows the time spent moving during a mission by all the robot team members combined. For both the simulated and physical runs, a pattern similar to that exhibited by execution time is noticed, which shows that the robots spend more time moving with increasing  $P$ . However, as noted in the previous metric, the variance for the physical runs is much greater than that of the simulated runs.



**Fig. 2.** Results for *total movement time*, with increasing  $P$

The **total distance travelled** metric is the total distance travelled by all the robot team members combined. The trend is identical to execution time and total movement time.

The **total collisions** metric counts the number of times that the robots either collided or came close to each other and had to execute a collision avoidance behaviour. This metric exhibits unexpected results, as shown in Fig. 3. The results have identical patterns for both simulated and physical runs (discussed further in Sect. 4). The increase between PL-0 and PL-25 is expected, though it is unexpected for collisions to decrease for  $P$  greater than PL-25. This outcome has been identified to result from the high amount of messages being dropped by the function in the code which updates the pose of consequent robot-team members. Note that, for purposes of these experiments, the robot controller does not record collisions detected by the robot’s bump sensors.



**Fig. 3.** Results for number of *near collisions*, with increasing  $P$

The performance metrics **total delay time** and **total idle time** are not as relevant for testing communication quality directly, but still reflect aspects of team performance. Delay time is related to the total “near collisions”. The

MRTeAm framework employs a simplistic collision avoidance behaviour: when two robots come within a pre-defined  $\epsilon$  of each other, they both stop; the robot that is closest to its next task location is given right-of-way, and the other robot waits until the first robot has cleared its path. The amount of time that a robot stops in order to let another robot pass is called “delay time”. The second metric, idle time, is related to the efficiency with which the team collectively completes all the tasks assigned to the them. When a robot finishes executing its assigned tasks, it waits until all the other robots on the team have finished executing their assigned tasks. That amount of waiting time is called “idle time”.

## 4 Analysis of Results

In this section, we describe our statistical analysis of the experimental results obtained and presented in the previous section. Our aim is to show that there are statistically significant differences between the different levels of packet loss. First we test the distribution of the raw data for normality, to determine whether our data sets are parametric or non-parametric. Then we run t-tests to evaluate for statistical significance.

### 4.1 Shapiro-Wilk Test

The Shapiro-Wilk test is used to show that the data samples obtained from the experiment are more likely to have a normal distribution. The algorithm used for the Shapiro-Wilk test is from *scipy.stats.shapiro* [1].

Shapiro-Wilk test is highly recommended for sample sizes less than 50. For a sample size of  $\leq 25$ , the Shapiro-Wilk value (W) needs to be in the range 0.918–0.989 [14], and for a sample size of  $\leq 20$ , the W value needs to be in the

**Table 1.** Shapiro-Wilk values for *execution time*.

(a) Simulation	
Parameter	Shapiro-Wilk (W)
PL-0	0.974
PL-25	0.979
PL-50	0.983
PL-75	0.956
(b) Physical	
Parameter	Shapiro-Wilk (W)
PL-0	0.606
PL-25	0.746
PL-50	0.718
PL-75	0.799

**Table 2.** Shapiro-Wilk values for *total movement time*.

(a) Simulation	
Parameter	Shapiro-Wilk (W)
PL-0	0.892
PL-25	0.926
PL-50	0.876
PL-75	0.907
(b) Physical	
Parameter	Shapiro-Wilk (W)
PL-0	0.868
PL-25	0.708
PL-50	0.836
PL-75	0.930

**Table 3.** Shapiro-Wilk values for *distance travelled*.

(a) Simulation	
Parameter	Shapiro-Wilk (W)
PL-0	0.967
PL-25	0.557
PL-50	0.917
PL-75	0.974
(b) Physical	
Parameter	Shapiro-Wilk (W)
PL-0	0.724
PL-25	0.634
PL-50	0.842
PL-75	0.833

range 0.905–0.988 [14]. However, since the test is very biased on sample size and is only hypothesised, we additionally examined quantile-quantile (Q-Q) plots to visually verify the normality of the distribution. Since the Q-Q plots are used as a visual aid we did not include these in the Analysis Section. The “near collisions” metric returns non-parametric data. Therefore, it needs to be interpreted in a different fashion compared to the other metrics. Moreover, the results actually show an increase in near collisions with  $P$  equal to PL-25, but a sudden decrease for  $P$  greater-than-equal to PL-50, which is an unexpected decrease due to the individual robots’ navigation dropping localisation messages. Below, Tables 1, 2 and 3 show the performance metrics given a packet-loss parameter, which

are expected to be parametric. The simulation results for the  $W$  value indicate that the data is normally distributed, whereas the physical results initially show similar properties but with much higher variance in the data (primarily because there is noise in communication and localisation in the physical environment).

## 4.2 Paired t-test

The paired t-test is used to analyse the experimental data and test for the null hypothesis ( $H_0$ ). The  $H_0$  tests if, for a given metric, two different and independent packet-loss parameters return identical means. The method *scipy.stats.ttest\_ind* [1] is used to perform a standard independent paired t-test calculation. In this experiment, the alternate hypothesis ( $H_A$ ) is accepted if the significance level is 5% or less, which means that there is 95% probability that the results achieved did not occur by chance.

In the general case for the simulation runs, the alternative ( $H_A$ ) hypothesis is accepted for the performance metrics for the highest subsidiary parameter (PL-75). This implies that, for the tested performance metrics, if  $P$  is greatly increased, it shows significant reduction in performance for the multi-robot team. This result is not observed across all the performance metrics tested in the physical runs. We believe that several factors contribute to this result: the scenario ( $S_2$ ) is

**Table 4.** The table shows the t-test, p-value and the hypothesis decision for *execution time* for the simulation runs.

Primary parameter	Subsidiary parameter	T	p	Hypothesis ( $H_0/H_A$ )
PL-0	PL-25	-3.63	0.00	$H_A$
	PL-50	-3.86	0.00	$H_A$
	PL-75	-3.87	0.00	$H_A$
PL-25	PL-50	0.03	0.98	$H_0$
	PL-75	-1.18	0.27	$H_0$
PL-50	PL-75	-1.03	0.31	$H_0$

**Table 5.** The table shows the t-test, p-value and the hypothesis decision for *execution time* for the physical runs.

Primary parameter	Subsidiary parameter	T	p	Hypothesis ( $H_0/H_A$ )
PL-0	PL-25	-1.22	0.23	$H_0$
	PL-50	-1.83	0.07	$H_0$
	PL-75	-2.55	0.01	$H_A$
PL-25	PL-50	-0.94	0.36	$H_0$
	PL-75	-1.66	0.10	$H_0$
PL-50	PL-75	-0.67	0.51	$H_0$



**Table 6.** The table shows the t-test, p-value and the hypothesis decision for *total movement time* for the simulation runs.

Primary parameter	Subsidiary parameter	T	p	Hypothesis ( $H_0/H_A$ )
PL-0	PL-25	-0.63	0.53	$H_0$
	PL-50	-1.86	0.07	$H_0$
	PL-75	-4.79	0.00	$H_A$
PL-25	PL-50	-1.19	0.24	$H_0$
	PL-75	-3.97	0.00	$H_A$
PL-50	PL-75	-2.50	0.02	$H_A$

**Table 7.** The table shows the t-test, p-value and the hypothesis decision for *total movement time* for the physical runs.

Primary parameter	Subsidiary parameter	T	p	Hypothesis ( $H_0/H_A$ )
PL-0	PL-25	-0.21	0.83	$H_0$
	PL-50	-0.99	0.33	$H_0$
	PL-75	-1.67	0.10	$H_0$
PL-25	PL-50	-0.63	0.53	$H_0$
	PL-75	-1.14	0.26	$H_0$
PL-50	PL-75	-0.48	0.63	$H_0$

**Table 8.** The table shows the t-test, p-value and the hypothesis decision for *total distance* for the simulation runs.

Primary parameter	Subsidiary parameter	T	p	Hypothesis ( $H_0/H_A$ )
PL-0	PL-25	-0.33	0.75	$H_0$
	PL-50	1.07	0.23	$H_0$
	PL-75	2.03	0.05	$H_A$
PL-25	PL-50	0.98	0.33	$H_0$
	PL-75	1.59	0.12	$H_0$
PL-50	PL-75	0.75	0.46	$H_0$

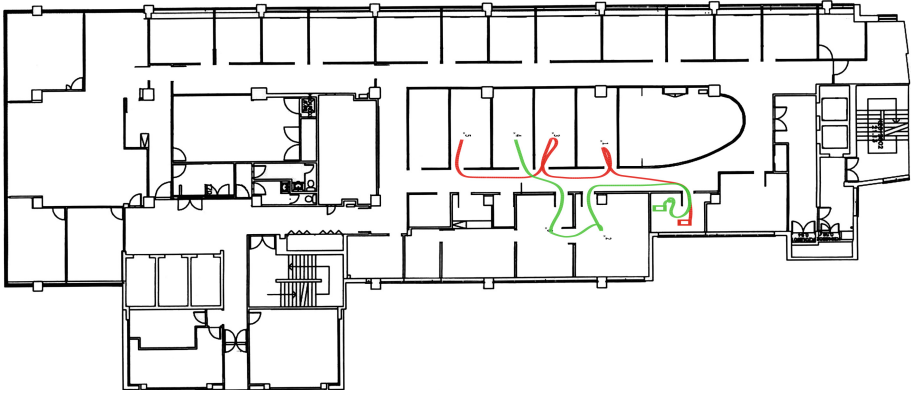
much simpler to navigate than scenario  $S_1$  and there is noise in the localisation for the physical robots (but not in the simulation). Similar to Sect. 3.1, two separate tables are used to show the hypothesis decision for each particular performance metric for the simulation and physical runs (Tables 4, 5, 6, 7, 8 and 9).

**Table 9.** The table shows the t-test, p-value and the hypothesis decision for *total distance* for the physical runs.

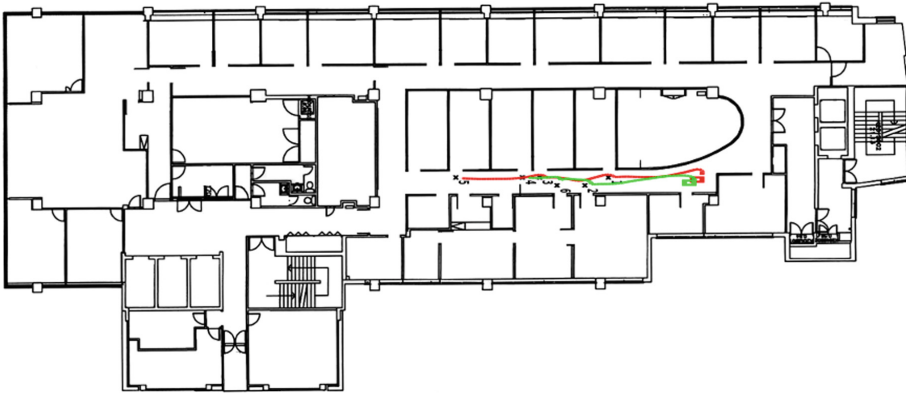
Primary parameter	Subsidiary parameter	T	p	Hypothesis ( $H_0/H_A$ )
PL-0	PL-25	-0.54	0.59	$H_0$
	PL-50	-1.90	0.06	$H_0$
	PL-75	-2.48	0.02	$H_A$
PL-25	PL-50	-1.10	0.28	$H_0$
	PL-75	-1.72	0.09	$H_0$
PL-50	PL-75	-0.75	0.50	$H_0$

### 4.3 Trajectories

The results of the trajectory representations are analysed to identify change in performance with change in  $P$ . Trajectory Fig. 4 represents a simulated run, showing scenario  $S_1$ , and trajectory Fig. 5 represents a physical run, showing scenario  $S_2$ . With the trajectory representations, it can clearly be seen why there are differences between the simulated and physical run results.



**Fig. 4.** Simulation run showing robot team trajectory in an office setting, with PL-0, and 2 robots (robot 1 shown in red and robot 2 shown in green). (Color figure online)



**Fig. 5.** Physical run showing robot team trajectory in an office setting, with PL-0, and 2 robots (robot 1 shown in red and robot 2 shown in green). (Color figure online)

## 5 Related Work

The primary motivation for this line of research was due to past experiences with multi-robot teams where there had been a variety of network connectivity problems. Multi-robot team communication is an important and complex issue when it comes to performing heterogeneous tasks (e.g. two or more robots working together to move heavy objects out of the way to reach a goal). However, communications have limitations and infrastructure may break down.

In works by Zadorozhny and Lewis [18] and Murphy *et al.* [8] the authors investigate different communication methods in the human-robot communication domain. An extra constraint exists on the human-robot system, which is that robots need to be equipped with specific sensors to communicate/transmit information to a human controller, as highlighted by Murphy *et al.* [8]. [8,18] mention the constraints experienced by communication limitations, but none of the aforementioned works focus on solving or quantifying this issue.

By widening our research, we found similar issues in the human-human communication domain. The researched works [2,15,17] all had the common goal of establishing a communication network in a disaster area, where a network would be fragmented or non-existent. The purpose of this was to then allow a “tethered” link between the general public and first responders (police/ambulance). In [2] the authors investigate ways to improve message passing and to create effective ad-hoc networks.

In [16], the authors examine possible formation strategies in simulations to assist and maintain communications, while multi-robot teams perform exploration and victim-locating tasks. Moreover, although [16] does not focus on reducing communication failure, they concentrate on providing improved communication for multi-robot teams. The research by Jensen *et al.* [6] employs their own *Sweep Exploration Algorithm (SEA)* for the coverage of unknown environments. The algorithm allows a multi-robot team to expand their exploration in a tree-like structure. A constraint is placed on the multi-robot team to always

maintain communication between members. The method used by Jensen *et al.* [6] introduces some weaknesses prone to failure, particularly in physical environments. Another work by Gunn and Anderson [5] describes a framework for multi-robot teams that allocates roles and allows robots to dynamically change roles depending on mission requirements or environmental conditions (e.g. to compensate for a lost team member or to complete a victim-locating task). The primary goal for the framework is to assist in team maintenance and task management, but the authors note that for large percentages of communication loss, the performance of all methodologies was poor. This was mainly due to a very high message failure rate, which resulted in failure to allocate tasks, and made team coherence impossible to maintain over time.

These findings are the source of our long-term motivation with respect to the eventual deployment of multi-robot teams to help first responders.

## 6 Conclusions and Future Work

We have presented results of a preliminary set of experiments that attempt to quantify the impact of packet loss on a multi-robot team. Two different scenarios were evaluated, one with a team of simulated robots and the other with a team of physical robots. The two scenarios ( $S_1$  and  $S_2$ ) are very different and thus limit the accuracy of the results for the simulation and physical runs. However, when examining the results of the simulation and physical runs, similar trends emerge specifically in the “execution time”, “total movement time”, “total distance travelled” and the “near collisions” metrics. The results obtained give a starting baseline for how communication quality affects multi-robot teams. However, the experiments do not give conclusive insight on how degradation of communication quality affects the performance metrics that are tested. Unfortunately, it is difficult to run  $S_1$  with the physical robots because the locations of the tasks in scenario  $S_1$  are inside people’s offices, and we don’t necessarily have access to all those points. Instead, our immediate next step is to develop a third scenario  $S_3$  that adds complexity to the mission and can be evaluated both in simulation and with physical robots.

In future work, we will use an *ad-hoc* network (similarly to those described in [2, 8, 11]) to facilitate communication amongst robots and between the robots and the server. This will allow the robots to measure real signal strength and adapt their behaviour accordingly, so that they do not lose connectivity. We will investigate further the strategies proposed by Takahashi *et al.* [16] in a physical environment, and additionally employ certain aspects of the SEA algorithm by Jensen *et al.* [6].

## References

1. Python: Scipy stats library (2008–2016). <https://docs.scipy.org/doc/scipy-0.15.1/reference/stats.html>
2. Al-Akkad, A., Raffelsberger, C., Boden, A., Ramirez, L., Zimmermann, A.: Tweeting when online is off? opportunistically creating mobile ad-hoc networks in response to disrupted infrastructure. In: 11th International ISCRAM Conference Proceedings, Pennsylvania, May 2014

3. Gerkey, B., Vaughan, R.T., Howard, A.: The player/stage project: tools for multi-robot and distributed sensor systems. In: Proceedings of the 11th International Conference on Advanced Robotics (2003)
4. Gerkey, B.P., Mataric, M.J.: A formal analysis and taxonomy of task allocation in multi-robot systems. *Int. J. Robot. Res.* **23**(9), 939–954 (2004)
5. Gunn, T., Anderson, J.: Dynamic heterogeneous team formation for robotic urban search and rescue. *J. Comput. Syst. Sci.* **81**(3), 553–567 (2015)
6. Jensen, E.A., Nunes, E., Gini, M.: Communication-restricted exploration for robot teams. In: Proceedings of the 28th AAAI Conference on Artificial Intelligence (2014)
7. Landén, D., Heintz, F., Doherty, P.: Complex task allocation in mixed-initiative delegation: a UAV case study. In: Desai, N., Liu, A., Winikoff, M. (eds.) PRIMA 2010. LNCS, vol. 7057, pp. 288–303. Springer, Heidelberg (2012). doi:[10.1007/978-3-642-25920-3\\_20](https://doi.org/10.1007/978-3-642-25920-3_20)
8. Murphy, R.R., Srinivasan, V., Henkel, Z., Suarez, J., Minson, M., Straus, J., Hempstead, S., Valdez, T., Egawa, S.: Interacting with trapped victims using robots. In: IEEE International Conference on Technologies for Homeland Security (HST), Waltham, Massachusetts, November 2016
9. Quigley, M., Conley, K., Gerkey, B.P., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: ROS: an open-source robot operating system. In: ICRA Workshop on Open Source Software (2009)
10. Rowan, D.: Kinect for Xbox 360: The inside story of Microsoft’s secret ‘Project Natal’. *Wired*, 29 October 2010
11. Ruiz-Zafra, A., Nez, A.G., Penads, M.C., H.Cans, J., Borges, M.R.: SUCRE: supporting users, controllers and responders in emergencies. In: 11th International ISCRAM Conference Proceedings, Pennsylvania, May 2014
12. Schneider, E., Sklar, E.I., Parsons, S.: Evaluating multi-robot teamwork in varied environments. In: Proceedings of the 17th Towards Autonomous Robotic Systems (TAROS), awarded Best Student Paper Prize (2016)
13. Schneider, E., Sklar, E.I., Parsons, S., Özgelen, A.T.: Auction-based task allocation for multi-robot teams in dynamic environments. In: Dixon, C., Tuyls, K. (eds.) TAROS 2015. LNCS, vol. 9287, pp. 246–257. Springer, Cham (2015). doi:[10.1007/978-3-319-22416-9\\_29](https://doi.org/10.1007/978-3-319-22416-9_29)
14. Shapiro, S.S., Wilk, M.B.: An analysis of variance test for normality. *Biometrika* **44**(3/4), 591–611 (1995)
15. Sutton, J., Spiro, E., Fitzhugh, S., Johnson, B., Gibson, B., Butts, C.T.: Online message amplification in the boston bombing response. In: 11th International ISCRAM Conference Proceedings, Pennsylvania, May 2014
16. Takahashi, T., Kitamura, Y., Miwa, H.: Organizing rescue agents using ad-hoc networks. In: Pérez, J., et al. (eds.) Highlights on Practical Applications of Agents and Multi-Agent Systems. AISC, vol. 156, pp. 139–146. Springer, Heidelberg (2012). doi:[10.1007/978-3-642-28762-6\\_17](https://doi.org/10.1007/978-3-642-28762-6_17)
17. Tapia, A.H., LaLone, N., Kim, H.W.: Run Amok: group crowd participation in identifying the bomb and bomber from the boston marathon bombing. In: 11th International ISCRAM Conference Proceedings, Pennsylvania, May 2014
18. Zadorozhny, V., Lewis, M.: Information fusion for USAR operations based on crowdsourcing. In: 16th International Conference on Information Fusion (FUSION), Istanbul, Turkey, July 2013