




# MRCComm: Multi-Robot Communication Testbed

Tsvetan Zhivkov, Eric Schneider<sup>(✉)</sup>, and Elizabeth Sklar<sup>(✉)</sup> 

Department of Informatics, King's College London, London, UK  
{tsvetan.zhivkov,eric.schneider,elizabeth.sklar}@kcl.ac.uk  
<http://www.kcl.ac.uk>

**Abstract.** This work demonstrates how dynamic robot behaviour that responds to different types of network disturbances can improve communication and mission performance in a *Multi-Robot Team (MRT)*. A series of experiments are conducted which show how two different network perturbations (i.e. packet loss and signal loss) and two different network types (i.e. wireless local area network and ad-hoc network) impact communication. Performance is compared using two MRT behaviours: a baseline versus a novel dynamic behaviour that adapts to fluctuations in communication quality. Experiments are carried out on a known map with tasks assigned to a robot team at the start of a mission. During each experiment, a number of performance metrics are recorded. A novel dynamic *Leader-Follower (LF)* behaviour enables continuous communication through two key functions: the first reacts to the network type by using signal strength to determine if the robot team must commit to grouping together to maintain communication; and the second employs a special task status messaging function that guarantees a message is communicated successfully to the team members. The results presented in this work are significant for real-world multi-robot system applications that require continuous communication amongst team members.

**Keywords:** Multi-robot team · Behaviour-based control · Dynamic roles

## 1 Introduction

In a *Multi-Robot Team (MRT)*, providing correct and current information to team members are two of the critical functions that depend on networked communication facilities. Reliable communication is such an important aspect in robotics that it prompted a fundamental change in the communication middleware used in the Robot Operating System (ROS1) [13] from *Publisher-Subscriber (Pub-Sub)* to the open source *Data Distributed Service (DDS)* that is being integrated into ROS2 [5, 6]. As home automation and mobile technologies grow, the

---

This work was partially supported by a King's College London Graduate Training Scholarship and by the ESRC under grant #ES/P011160/1.

potential applications for MRTs also expand [1, 11, 19]. Furthermore, robots are often deployed in mission scenarios that are unsuitable or dangerous for human operators and which often have poor or crippled network infrastructure, such as urban search-and-rescue (USAR) [12, 18, 19], humanitarian de-mining or nuclear plant monitoring. Even state-of-the-art deployment of telecommunication networks and research that addresses latency and outage issues may experience poor routing, network congestion, channel interference or packet dropping, which can have significant impact on robot systems that rely on timely and accurate mission critical information, as noted by Caccamo [4] and Kashino *et al.* [9].

In earlier work [20], we employed the ROS-based *MRTeAm* (*Multi-Robot Task Allocation*) framework [15] as the basis for a study in which we applied a probabilistic message loss function to one message “topic”<sup>1</sup> that is shared amongst the robot team members. The affected topic, *AmclPose*<sup>2</sup>, comprises messages that receive and send data about a robot’s position in a known map. We ran a series of experiments to measure the impact on mission performance metrics when *AmclPose* messages were lost at increasingly frequent rates. Our results showed non-linear degradation in performance as message loss probability grew from 0% to 75%. Although limited, these results gave us an initial understanding of how a multi-robot team is affected by lowering communication quality. Here, the probabilistic message loss function is applied to two message topics, *AmclPose* and *TaskStatus*. Moreover, we expand the experiment configuration to include network types, network perturbations, new performance metrics, message functionality and behaviours. We demonstrate experimentally the range of effects that various network perturbations have on multiple aspects of team performance and how this changes with different network types. To facilitate our empirical investigation, we have developed the **MRCComm** (Multi-Robot Communication) testbed that allows for control of communication for individual message topics and thus subsequently experimental analysis by topic. Though the results here are for all message topics, MRCComm makes use of a novel, dynamic *Leader-Follower* (*LF*) behaviour inspired by the concept of infrastructure-less (i.e. ad-hoc) networks, which is used to respond to real-time fluctuations in network connectivity. Moreover, we employ a novel messaging function that does not require any changes to the underlying pub-sub communication middleware while offering a best effort to verify acknowledgement of message transmission. Our results show that the LF behaviour and the new message function maintain continuous communication regardless of the network type and network perturbation that effects the communication quality. This is a crucial step for multi-robot research toward acquiring the tool set needed to assess and adapt to unreliable communication and maintain continuous connectivity. Our long-term aim is to improve message passing capabilities in MRTs, by providing adaptive behaviours that respond to different network problems which arise during a mission.

---

<sup>1</sup> A “topic” in a Publisher-Subscriber (Pub-Sub) communication system refers to a category of related messages that are defined and grouped together when the applied system is engineered.

<sup>2</sup> Adaptive Monte Carlo localisation (position estimation) for mobile robots [7].

The remainder of the paper is structured as follows. In Sect. 2, we review related work. Section 3 briefly describes the approach, the architecture of our system and expansion of our framework, MRComm. Section 4 outlines the experimental setup for performing the set of experiments, which are designed to analyse performance of communication between the baseline and novel behaviours. In Sect. 5 we present our experimental results and discussion. Finally, we close in Sect. 6 with a brief summary and directions for ongoing and future research.

## 2 Related Work

The motivation to analyse, react and mitigate the effects of degrading communication quality in MRTs started in the mobile device domain. Although research on communication networks in the mobile device domain is plentiful, this is not the case for the MRT domain. However, an overlap exists between these domains as shown by Witkowski *et al.* [18] and Lujak *et al.* [11]. These works investigate different outcomes but use similar methods for communication, i.e. *mobile ad-hoc network* (MANET) or leverage smart devices for communication. When looking at research on the effects of communication networks in only the Robot and MRT domains, it is clear that it is still in its early stages and there are many aspects to be still considered. In work by Murphy *et al.* [12] a remote controlled robot is used to perform triage on a victim in a search-and-rescue scenario and they examine the impact of different sensors on communication (e.g., audio and video). Zadorozhny and Lewis [19] look at autonomous MRT collaboration with human assistants to perform search and rescue of victims in a simulated environment. The work by Kashino *et al.* [9] looks at optimal predetermined delivery of static-sensor networks using MRTs to cover an area to enable complete communication. This work shows motivation from the need to create network infrastructure in an infrastructure-less environment. The notion of using ad-hoc networks for communication in multi-robot systems isn't thoroughly covered. However, some works such as Takahashi *et al.* [17] investigate, in simulation, MRT formations with the aim of using an ad-hoc network. Furthermore, Witkowski *et al.* [18] looks at reestablishing infrastructure using robot teams and ad-hoc networks in disaster zones. Finally, Caccamo *et al.* [4] demonstrate a novel robot navigation planner, in simulation, that is communication-aware. We look at one of the initial works on behaviour based control for MRTs by Balch and Arkin [3]. Their work is focused on the interaction among lower level systems (e.g., navigation and obstacle avoidance) and formation control, and on analysing the strengths and weaknesses of different formation patterns; however, it is not inspired by MRT communication. Although we draw insight from [3], our behaviour based on the leader-follower paradigm does not directly interact with the lower level systems or adopt any particular formation control, which is explained in *Robot Behaviours* Sect. 3.1. We combine the analysis of communication issues of shared messages between robots, different network parameters and the use of behaviour-based control of MRTs into one testbed, MRComm, which we present here. The ROS platform is originally designed for single robot

academic experiments, with no real-time requirements and an assumption that wireless local area network connectivity is available and good. Research and real use-cases now extend the use of MRTs into a number of different environments where connectivity is poor or no network infrastructure exists at all. However, while it is possible to create multi-robot systems using ROS, there is no standardised approach. Moreover, there are works such as [2] and MRTeAm [15] that provide the tools to create MRS.

### 3 Approach

Our overall line of work on multi-robot teams examines various problems related to *coordination*, with the ultimate goal of developing strategies that guarantee efficient and effective mission completion. We have produced a number of metrics that capture detailed aspects of team performance, these are discussed further in Sect. 5. The contribution described here builds upon this exploration of the multi-robot team coordination domain and specifically investigates the importance of reliable communication within this domain. While our earlier work studied the impact of different market-based mechanisms to distribute tasks amongst team members [15], in the setup employed here, messages are passed which: (1) directly assign tasks to robots instantaneously and sequentially; (2) provide location information about robots' positions, as input to the task distribution process and to facilitate collision-free movement; and (3) report task completion status, possibly accompanied by sensor data acquired as part of the task. The robots are given tasks by an assigner agent (i.e., robot, remote or virtual agent) which initiates messages of topic 1, and the assigned robots initiate the other message topics (2 and 3). Our previous investigation into the impact of poor communication in multi-robot teams only considered failure of message topic 2. Here, we consider failure of message topics 2 and 3, which constitutes *AmclPose* (team position messages) and *TaskStatus* respectively.

#### 3.1 MRComm Testbed

Here, we describe our MRComm testbed, which is built on MRTeAm, the software framework mentioned earlier which we designed for conducting research on multi-robot task allocation [14–16]. Both layers rely on ROS [13] and employ two main types of components: a centralised agent that distributes tasks to robots and multiple robot controller agents for executing the tasks. Furthermore, our simulated experiments are conducted in the mobile robot simulator Stage<sup>3</sup>. In MRComm, the “auctioneer” is replaced by an *assigner* agent, as we shift our research emphasis from task allocation (in MRTeAm) to team communication (in MRComm)—assigning tasks directly to robots using a fixed distribution that is defined *a priori* as part of a mission configuration. The robot controller agent is extended as discussed below, to be able to respond dynamically when communication problems arise. The assigner agent used in MRComm is responsible for

<sup>3</sup> <http://wiki.ros.org/stage>.

loading a mission configuration and assigning tasks to all team members sequentially. The assigner also acts as a recording agent, without interfering, recording received experiment and team messages. The MRComm testbed defines a failed task when the recording agent does *not* receive a SUCCESS message after a mission has been completed. The robot controller is initialised with parameters for behaviour, scenario, network perturbation and network type at the start of an experiment. Thereafter, it receives tasks from the assigner agent and begins to execute them.

**Network Type.** The network type is the communication network used in experiments: *WiFi* via either a wireless local area network (WLAN) or an *ad-hoc* (AH) network<sup>4</sup>. To create the AH network, devices connect directly to one robot and rely on the close proximity of neighbouring devices to maintain connectivity. Devices can also leave and join the network freely without issues; however, shared information is only available as long as connections are maintained. The characteristics of the AH network are: no infrastructure, quick dissemination of information and distributed control (i.e., no single point of failure). We impose network limitations to make our problem tractable by assuming specific WLAN and AH network conditions. For the simulation experiments presented here, we modelled the limitations of our ad-hoc network using Turtlebot2 robots<sup>5</sup> and the type of IEEE 802.11n/ac wireless network cards that come standard with that platform. We measure the signal strength at a high resolution and take over thirty readings per resolution in order to construct a realistic model for our experiments, as shown in Fig. 1. From Fig. 1, we conclude that the AH network limit for communication is  $\approx 8.0$  m. After this limit, the signal becomes oversaturated or too weak and as a result drops consistently below  $-70$  dBm, which makes predicting distance impossible. Moreover, for both WLAN and AH, it is assumed that *signal-to-noise-ratio* (SNR) experiences uniform loss and SNR interference from other devices (not our robots) is negligible. Additionally for WLAN, we assume uniform radial coverage of the operational environment.

**Network Perturbations.** In our experiments, we apply a network perturbation mechanism to disrupt the quality of communication. We analyse the effects on team performance of two such mechanisms: *simulated packet-loss* (SPL) and *simulated signal loss threshold* (SLT). The SPL mechanism impacts communication quality by dropping a certain percentage, such as  $\{0\%, 25\%, 50\%, 75\%\}$ , of the shared messages (i.e., topics 2 and 3, as mentioned above). The SLT mechanism shows the effect that limited signal strength has on the MRT. The threshold distance used for SLT is 6.0 m. The SLT mechanism is only employed in experiments with the AH network; given our assumptions, made above, about the WLAN network coverage hold, the implementation of SLT within a WLAN environment is meaningless.

**Robot Behaviours.** We compare two different robot behaviours: a baseline *no-behaviour* (NB) and our novel *Leader-Follower* (LF) behaviour, which is designed

<sup>4</sup> Currently the ad-hoc network is simulated.

<sup>5</sup> <https://www.turtlebot.com/turtlebot2/>.

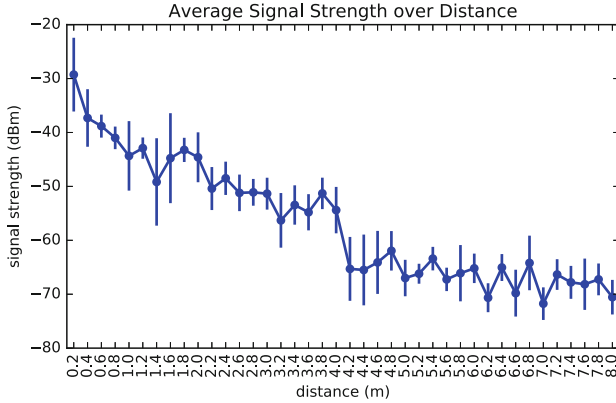


Fig. 1. Signal strength vs distance. Average values over 30 readings.

to respond to and maintain communication regardless of network type or perturbation. In NB mode, robot team members do not adjust their behaviour based on network quality. They attempt to complete their assigned tasks, disregarding network parameters or loss of communication, and perform standard navigation and obstacle avoidance behaviours. The LF behaviour is inspired by the AH network type, in which change in signal strength (communication quality), modelled as a function of distance, is detected as the robot team move away from each other, triggering the action of “regrouping” to maintain communication. In order to regroup, LF has its own signal strength threshold limit, which is approximately 5.0 m as depicted in Fig. 4(b). In LF mode, no experiments are executed using the WLAN network type; as we expect our complete and uniform radial coverage assumption to hold. The action of regrouping can be translated easily to react to dynamic change in network type as well, for example from WLAN to AH and back again.

When the robot agents use LF behaviour, they assume one of three roles: *not assigned (NA)*, *leader* or *follower*. Initially all robots start with the NA role. Upon the team detecting a loss of connection from any member, the robots dynamically assign themselves to either the *leader* or *follower* role, based on a The utility score,  $u$ , is defined as follows:

$$u = d\_score * num\_incomplete * recently\_completed$$

where:

- $d\_score$  = distance score, computed as  $1/distance\_to\_goal$  (task location);
- $num\_incomplete$  = number of incomplete tasks remaining on the robot’s *agenda*<sup>6</sup>, which is computed as the total number of tasks assigned less the number of tasks completed;

<sup>6</sup> The “agenda” is the list of tasks a robot has been allocated by the assigner agent.

- *recently\_completed* = 0.5 if the robot has just completed a task or 1.0 if it has not (this value is reset with every change in role and/or completion of a task).

This last factor acts to balance out the priorities of tasks amongst the teammates. This is because the *follower* behaviour prioritises staying in communication with teammates over completing its allocated tasks, whereas the *leader* robot prioritises completing its tasks. In effect it prevents a deadlock in roles from occurring, for example having the same robot as leader. Effectively, this factor ensures that all tasks are given priority at some point during the mission. The robot with the highest  $u$  value is selected as *leader*. In our simulation, the leader is a proxy for the robot that initialises the ad-hoc network in a physical setup. Then the followers connect to this new network. The final stage of the behaviour clears all robots of their roles, i.e., NA, which we denote as *switching*.

The switching behaviour helps mitigate communication loss when using the AH network and the SLT network perturbation. The unique message function, implemented in LF, helps mitigate communication loss when using the AH network with the SPL network perturbation. The rationale for using our message function over other communication methods is because *TaskStatus* messages are of light load, do not require internal processing and can easily be analysed for communication quality. A status message sent using the message function includes a Boolean value, which is initially set to false. Once a robot sends a status message, it and any other robot that receives the message, will periodically re-send it. This continues until each robot *knows* that everyone in the team has received the message. This process is achieved by checking that the total number of robots that have re-sent the message is equal to the size of the team, which implies that all robots have received the message. The final step of the message function is to set the Boolean value to true and re-send the message, as illustrated in Fig. 2.

## 4 Experiments

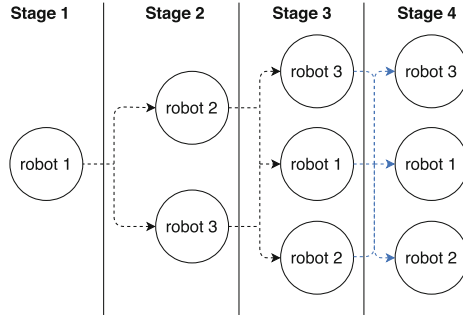
The experiments are defined as:

$$F_i = \{WLAN, AH\} \times \{SPL, SLT\} \times \{S_x\} \times \{NB, LF\}$$

where

- $F_i$  is an experiment setup with  $i \in \mathbb{N}$ ;
- the network types *WLAN* and *AH* represent wireless local area network (standard infrastructure) and ad-hoc network (no infrastructure) respectively (details described earlier);
- network perturbation *SPL* is simulated packet-loss where  $\{SPL0, SPL25, SPL50, SPL75\}$  denote  $\{0, 25, 50, 75\}$  percent of messages that are dropped respectively<sup>7</sup>;

<sup>7</sup> The 2 message topics effected by *SPL* are *TaskStatus* and team members' *AmclPose*.



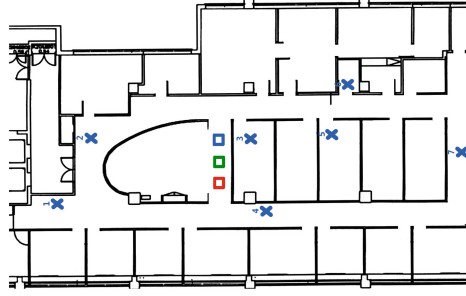
**Fig. 2.** Message propagation is shown by dotted lines in the diagram. In stage 1 a status message is sent; in stage 2 all team members have received the message; in stage 3 the entire team *knows* that the message has been received, Boolean value is set to true (i.e., this is further indicated by the blue dotted lines between stage 3 and 4); in stage 4 the message is sent with Boolean set to true and communication ends successfully. (Color figure online)

- $S_x$  is a task scenario where  $x \in \mathbb{R}^n$  is associated to specifically defined scenario containing sub-parameters  $n$  and  $m$ , which refer to the size of the team and number of tasks, respectively, described in [8, 10, 15, 16]; and
- the robot behaviours  $NB$  and  $LF$  denote our standard no-behaviour and our leader-follower behaviour, respectively.

For our experiment scenario, we have chosen 3 robots to perform 7 exploration tasks starting in a clustered formation, where each task is independent from the next and requires a single robot to complete it. We have purposefully chosen difficult task locations in narrow spaces and poor starting locations for the robot team (illustrated in Fig. 3). Tasks  $T_R$  are assigned to each robot  $R$  (see Fig. 3), and the assignments are fixed for all our experiments. The legend in Fig. 4 Sect. 5, consists of three tables that list the set of experiment configurations. For WLAN, we compared the four different SPL network parameters. For AH, in addition to the four SPL network parameters, we also compare SLT.

Each experiment is performed 30 times. We collect a number of different metrics during each experiment. The most relevant metrics discussed here are: *number of successful tasks, distance travelled, movement time, minimum and maximum separation distance, overall near collisions* and *idle time*. We expect that the number of successful tasks will decrease when the network is perturbed or when the network type is AH, except when employing the LF behaviour, which attempts to maintain connectivity. However, we expect an increase in distance travelled, time spent moving and overall near collisions by robots with the LF behaviour. The LF’s action of assigning roles and regrouping means that the robots are always busy moving and relatively close to each other in order to remain connected, which causes an increase in these three metrics. We predict that minimum and maximum separation distance among team members will be very small and the time spent idle after robots are done with their agenda is going to be reduced with LF compared to NB.

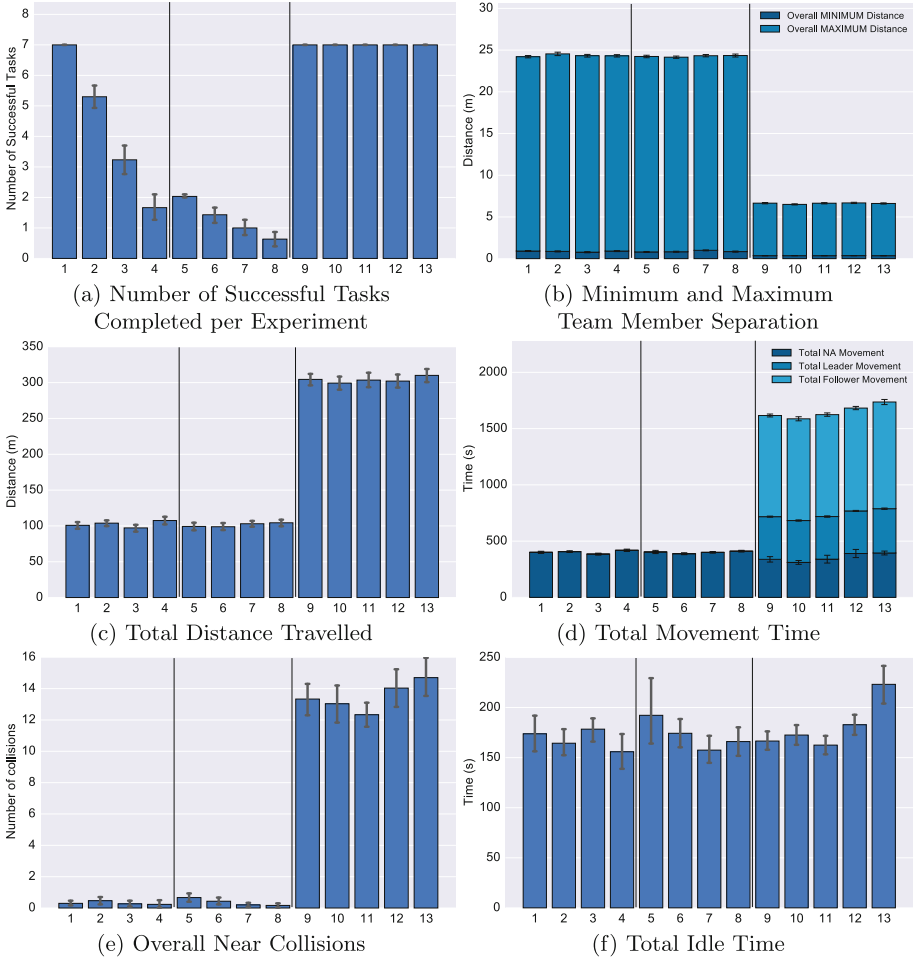




**Fig. 3.** Office setting for experiments, crosses represent task locations and squares represent robots (based on actual floor plan of building). Robot\_1 (red square) is assigned tasks  $T_1 = \{1, 4, 7\}$ , robot\_2 (green square)  $T_2 = \{2, 5\}$  and robot\_3 (blue)  $T_3 = \{3, 6\}$ . (Color figure online)

## 5 Results and Discussion

The legend in Fig. 4 is split into three tables in the same way as the resulting plots (a), (b), (c), (d), (e) and (f) to better highlight the changes in the performance metrics. The first set of results in plots (a) and (b) of Fig. 4 present the positive outcomes of using LF over NB for the MRT. Figure 4(a) presents the successful communication of task status messages for LF. However, NB increasingly fails to maintain successful communication as SPL increases and practically fails when the AH network type is used. Figure 4(b) demonstrates the minimum and maximum separation between team members throughout the duration of an experiment, which highlights an important dynamic between the two behaviours. As a result of LF's grouping capability, the minimum separation distance is approximately 0.35 m and the maximum is never greater than approximately 7.0 m. This is perfect for allowing communication when the AH network is used, although it is the primary reason for the increase that is observable in Figs. 4(c), (d) and (e). In NB mode, the minimum separation is approximately 0.40 m and the maximum is approximately 23.0 m. Thus connection fails in NB mode after the limits for communication, applied to AH and SLT (i.e., 8.0 m and 6.0 m, respectively), are reached. The next set of results in plots (c) and (d) in Fig. 4 demonstrate the distance travelled and the time spent moving by the robots to be about three times greater for LF compared to NB. This is the expected result due to LF's current design, which is depicted more clearly by the movement time plot, Fig. 4(d). Figure 4(d) shows the different design of LF's movement time, which is made up of three parts, namely NA, Leader and Follower movement time. For NB, movement time is made up of only NA movement time. The overall near collisions metric is much greater for LF than it is for NB. The reason for this is due to the grouping behaviour performed by LF, which is further established by Fig. 4(b). As the robots navigate the environment while performing grouping behaviour, in LF mode, the act of manoeuvring in close proximity creates a higher likelihood of a near collision, hence the performance seen in Fig. 4(e).



Label	Experiment Type
1	SPL0_NB_WLAN
2	SPL25_NB_WLAN
3	SPL50_NB_WLAN
4	SPL75_NB_WLAN

Label	Experiment Type
5	SPL0_NB_AH
6	SPL25_NB_AH
7	SPL50_NB_AH
8	SPL75_NB_AH

Label	Experiment Type
9	SLT_LF_AH
10	SPL0_LF_AH
11	SPL25_LF_AH
12	SPL50_LF_AH
13	SPL75_LF_AH

Legend

**Fig. 4.** Results show mean and standard deviation over 30 simulation trials for each experimental condition.

On the other hand, since NB performs no grouping, hence the MRT spread-out (i.e., maximum separation is very high), there is a lower likelihood of a near collision, i.e., Fig. 4(e). The idle time in Fig. 4(f), is the time accumulated after a robot has completed all the tasks in their agenda and is either waiting for the

rest of the team to complete their tasks or waiting to detect communication loss and is assigned a follower role. We expected that LF, in its current design, to perform worse and thus have an increased idle time. However, both behaviours achieved similar results in Fig. 4(f). Our results for SLT show very little impact on LF performance. As we expand SLT in future work, to be used with NB and WLAN, we expect this to change.

## 6 Summary

We have presented MRComm, a testbed, which utilises behaviours to deal with different network types and network perturbations. We present our results, in which certain performance metrics are used to evaluate how communication impacts MRT awareness and mission success. We show promising early results of our novel dynamic Leader-Follower behaviour and message function, which achieve perfect communication with a test set of network perturbations. The baseline MRT using only standard navigation and collision avoidance (NB behaviour) shows poor results in comparison. Our immediate next step is to demonstrate that MRComm can easily reproduce the same results in a physical environment. Furthermore, it is inevitable that in the real world, environments are dynamic and conditions change, including the type of network and perturbation. We wish to analyse how the LF behaviour can deal with dynamic network conditions. In future work, we will expand the network perturbation to simulated signal strength degradation and effective signal strength applied to physical robot experiments. We believe this will have a different impact on experiments using SLT and/or WLAN parameters. Finally, we hope to explore if other strategies improve the performance of the dynamic behaviour which can be particularly important for time-critical environments/missions such as search-and-rescue.

## References

1. 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. Theme 13 - Social Media in Crisis Response and Management, May 2014
2. Andre, T., Neuhold, D., Bettstetter, C.: Coordinated multi-robot exploration: out of the box packages for ROS. In: Published in IEEE Globecom Workshops (GC Wkshps), Austin, TX, USA (2014)
3. Balch, T., Arkin, R.C.: Behaviour-based formation control for multirobot teams. *IEEE Trans. Robot. Autom.* **14**, 926–939 (1998)
4. Caccamo, S., Parasuraman, R., Freda, L., Gianni, M., Ögren, P.: RCAMP: a resilient communication-aware motion planner for mobile robots with autonomous repair of wireless connectivity. In: International Conference on Intelligent Robots and Systems (IROS) (2017)
5. Dąbrowski, A., Kozik, R., Maciaś, M.: Evaluation of ROS2 communication layer. ROSCon, Seoul, Korea (2016)

6. Fernandez, E., Foote, T., Woodall, W., Thomas, D.: Next-generation ROS: building on DDS. ROSCon, Chicago, US (2014)
7. Fox, D., Burgard, W., Dellaert, F., Thrun, S.: Monte carlo localization: efficient position estimation for mobile robots. In: Published in Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI) (1999)
8. 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)
9. Kashino, Z., Nejat, G., Benhabib, B.: A multi-robot sensor-delivery planning strategy for static-sensor networks. In: International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada (2017)
10. 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 (LNAI), vol. 7057, pp. 288–303. Springer, Heidelberg (2012). [https://doi.org/10.1007/978-3-642-25920-3\\_20](https://doi.org/10.1007/978-3-642-25920-3_20)
11. Lujak, M., et al.: Towards robots-assisted ambient intelligence. In: Belardinelli, F., Argente, E. (eds.) EUMAS/AT -2017. LNCS (LNAI), vol. 10767, pp. 490–497. Springer, Cham (2018). [https://doi.org/10.1007/978-3-030-01713-2\\_34](https://doi.org/10.1007/978-3-030-01713-2_34)
12. Murphy, R., et al.: Interacting with trapped victims using robots. In: International Conference on Technologies for Homeland Security, HST (2013)
13. Quigley, M., et al.: ROS: an open-source robot operating system. In: ICRA Workshop on Open Source Software (2009)
14. Schneider, E., Sklar, E.I., Parsons, S.: Mechanism selection for multi-robot task allocation. In: Gao, Y., Fallah, S., Jin, Y., Lekakou, C. (eds.) TAROS 2017. LNCS (LNAI), vol. 10454, pp. 421–435. Springer, Cham (2017). [https://doi.org/10.1007/978-3-319-64107-2\\_33](https://doi.org/10.1007/978-3-319-64107-2_33)
15. Schneider, E., Sklar, E.I., Parsons, S.: Evaluating multi-robot teamwork in parameterised environments. In: Alboul, L., Damian, D., Aitken, J.M.M. (eds.) TAROS 2016. LNCS (LNAI), vol. 9716, pp. 301–313. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-40379-3\\_32](https://doi.org/10.1007/978-3-319-40379-3_32)
16. 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 (LNAI), vol. 9287, pp. 246–257. Springer, Cham (2015). [https://doi.org/10.1007/978-3-319-22416-9\\_29](https://doi.org/10.1007/978-3-319-22416-9_29)
17. 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. Advances in Intelligent and Soft Computing, vol. 156. Springer, Heidelberg (2012). [https://doi.org/10.1007/978-3-642-28762-6\\_17](https://doi.org/10.1007/978-3-642-28762-6_17)
18. Witkowski, U., et al.: Ad-hoc network communication infrastructure for multi-robot systems in disaster scenarios. In: Proceedings of IARP/EURON Workshop on Robotics for Risky Interventions and Environmental Surveillance, Benicassim, Spain (2008)
19. Zadorozhny, V., Lewis, M.: Information fusion for USAR operations based on crowdsourcing. In: Proceedings of the 16th International Conference on Information Fusion, Istanbul, Turkey (2013)
20. Zhivkov, T., Schneider, E., Sklar, E.I.: Measuring the effects of communication quality on multi-robot team performance. In: Gao, Y., Fallah, S., Jin, Y., Lekakou, C. (eds.) TAROS 2017. LNCS (LNAI), vol. 10454, pp. 408–420. Springer, Cham (2017). [https://doi.org/10.1007/978-3-319-64107-2\\_32](https://doi.org/10.1007/978-3-319-64107-2_32)