

Establishing Continuous Communication through Dynamic Team Behaviour Switching

Tsvetan Zhivkov¹, Eric Schneider², and Elizabeth I. Sklar³

Abstract— Maintaining continuous communication is an important factor that contributes to the success of multi-robot systems. Most research involving multi-robot teams is conducted in controlled laboratory settings, where continuous communication is assumed, typically because there is a wireless network (wifi) that keeps all the robots connected. But for multi-robot teams to operate successfully “in the wild”, it is crucial to consider how communication can be maintained when signals fail or robots move out of range. This paper presents a novel “leader-follower behaviour” with dynamic role switching and messaging that supports uninterrupted communication, regardless of network perturbations. A series of experiments were conducted in which it is shown how network perturbations effect performance, comparing a baseline with the new leader-follower behaviour. The experiments record metrics on team success, given the two conditions. These results are significant for real-world multi-robot systems applications that require continuous communication amongst team members.

I. INTRODUCTION

Continuous communication, operating in real-time and uninterrupted, is vital for a *multi-robot team (MRT)* to perform effectively. Providing correct information to team members and having up-to-date local knowledge are only two of the critical functions that depend on networked communication facilities. However, if and when network infrastructure breaks down, risking the loss of mission-critical messages, MRTs may be required to create opportunistic *ad-hoc (AH)* networks in order to sustain performance levels. The long-term vision for multi-robot systems communication is real-time, low-latency, zero-outage networks; but wide availability of such capabilities are far into the future. In the meantime, multi-robot systems research must develop strategies for overcoming network problems.

Here, we propose an approach to responding to poor network performance in a multi-robot team. In earlier work, we applied a probabilistic message loss function to assess the impact of dropped messages on team performance [1]. Although limited, this study gave us an initial understanding of how a multi-robot team is affected by degrading communication quality. In this paper, we take a step forward

in this line of research, by presenting a new MRT behaviour designed to adapt when communication fails and maintain better connectivity amongst team members.

We introduce *Leader-Follower (LF)* behaviour, described in Section III, a dynamic strategy that is inspired by the concept of AH networks. Experimental results show that our LF behaviour provides continuous communication regardless of network perturbations, as presented in Sections IV and V. This work takes a crucial step toward understanding how to assess and reduce the impact of unreliable communication depending on the network type and network perturbation that is experienced. 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.

II. RELATED WORK

We are not the first to investigate aspects of communication in multi-robot systems. Murphy *et al* [2] use a remote-controlled robot agent to perform triage on a victim in a search-and-rescue scenario and thoroughly examine the impact of different sensors on communication (e.g., audio and video). Zadorozhny and Lewis [3] look at autonomous MRT collaboration with human assistants to perform search and rescue of victims in a simulated environment. Lujak *et al* [4] propose a model for integrating multiple different technologies (e.g., mobile robots or mobile phones/devices) to assist victims during an emergency. These works highlight the importance that certain types of messages have, particularly in search-and-rescue and emergency-response situations, which has helped us to prioritise certain message types for experimentation, as described in Section III.

Furthermore, we explore works that use the notion of *ad-hoc* networks for communication in multi-robot systems. Takahashi *et al* [5] investigate, in simulation, MRT formations with the aim to use an ad-hoc network. Witkowski *et al* [6] look at reestablishing infrastructure using robot teams and *ad-hoc* networks. Caccamo *et al* [7] demonstrate a novel robot navigation planner in an urban search-and-rescue (USAR) simulation environment that is communication-aware and can repair lost communication.

The first generation *Robot Operating System (ROS1)* [8] platform was originally designed for single robot academic experiments, with no real-time performance requirements and an assumption that wireless local area network connectivity is good. In the next generation, ROS2 [9][10] the communication middleware has been updated to support real-time messaging. Recent work within the MRT research community has produced a few ROS-based frameworks and tools for experimenting with multi-robot systems [11][12],

¹Tsvetan Zhivkov is a Doctoral Student with the Department of Informatics Centre for Robotics Research (CoRe), King's College London University, Strand, London WC2R 2LS, England, United Kingdom.

tsvetan.zhivkov@kcl.ac.uk

²Eric Schneider is a Research Assistant with the Department of Informatics Centre for Robotics Research (CoRe), King's College London University, Strand, London WC2R 2LS, England, United Kingdom.

eric.schneider@kcl.ac.uk

³Elizabeth I. Sklar is a Professor of Robotics with the Department of Informatics, King's College London University, Strand, London WC2R 2LS, England, United Kingdom.

elizabeth.sklar@kcl.ac.uk

although there is no widely-adopted standard approach to facilitating multi-robot systems using ROS. As described below, we build on the MRTeAm framework [12] to implement and evaluate our Leader-Follower strategy.

III. APPROACH

We simulate *missions* where multi-robot teams are given a number of *tasks* to complete. Each task definition includes a location where the robot performs actions, such as *sensor-sweep* (collecting a series of images). In order to coordinate team activity, a centralised *assigner* agent determines which robots should perform which tasks and then mission execution begins. Thus the following types of messages occur:

- 1) the assigner agent sends messages that allocate tasks to robots;
- 2) the robots send messages to their teammates providing position information, which is used by the assigner for the task allocation process and by other robots to facilitate collision-free movement; and
- 3) the robots report task completion status, possibly accompanied by sensor data acquired as part of the task.

Our goal is to minimise the impact on team performance when some of these messages are not communicated.

A. Network type

We employ two different types of networks for our experiments: a baseline *wireless local area network (WLAN)*, which uses our local “wifi”, and an *ad-hoc (AH)* network.

We establish an *ad-hoc* radio communications network for our multi-robot system with the objective of maintaining continuous communication while performing normal operation. To create an AH network topology, devices connect directly to each other and rely on the close proximity of neighbouring devices to maintain connectivity. Devices can also leave and join the network freely; however, sharing of information is only possible as long as connections are maintained. The characteristics of our AH network are: no infrastructure, quick dissemination of information and distributed control (i.e., no single point of failure).

Radio signals have *theoretical* and *actual* limits. For our simulation, we measured the limitations of our *ad-hoc* network using Turtlebot2 robots and the type of IEEE 802.11n/ac wireless network cards which come standard with that platform. We measured the signal strength over various distances in order to construct a realistic model for our experiments, shown in Fig. 1. We employ these values in our LF behaviour, described below, as a guideline for maximum separation between any two robots in the team.

We impose further network limitations to make our problem tractable by assuming specific WLAN and AH network conditions: for both WLAN and AH it is assumed that SNR (signal-to-noise-ratio) experiences uniform loss and that interference from other devices is negligible, and additionally for WLAN we assume uniform radial coverage of the operational environment.

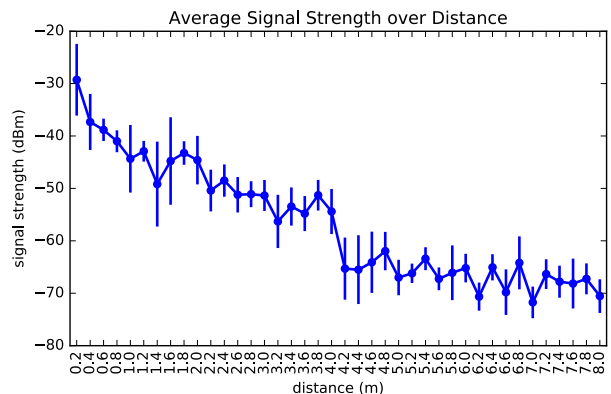


Figure 1. Signal strength vs. distance. Average values over 30 readings.

B. Robot behaviours

We compare two different robot behaviours: a baseline *no-behaviour (NB)* and our novel *Leader-Follower (LF)* behaviour, which is designed to maintain connectivity even when the network is unreliable. In NB mode, robot team members do not adjust their behaviour based on network quality. They attempt to complete their assigned tasks, disregarding network type or loss of communication quality, and perform standard navigation and obstacle avoidance behaviours.

In LF mode, robot team members detect changes in communication quality, such as when team members move outside of their close neighbourhood, requiring the team to regroup and move closer together again. This can be translated easily to react to 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, they all 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 utility score, defined as follows:

$$u = d_score \times num_incomplete \times recently_completed$$

where u = utility score; d_score = distance score, computed as $1 \div distance_to_goal$ (task location); $num_incomplete$ = number of incomplete tasks remaining on the robot's *agenda*¹, which is computed as the total number of tasks assigned less the number of tasks completed; $recently_completed$ = 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. If a given robot is always a follower, it may never get an opportunity to prioritise completion of its tasks. Effectively, this factor ensures that all tasks are given priority at some point during the mission.

¹ The “agenda” is the list of tasks a robot has been allocated by the assigner agent.

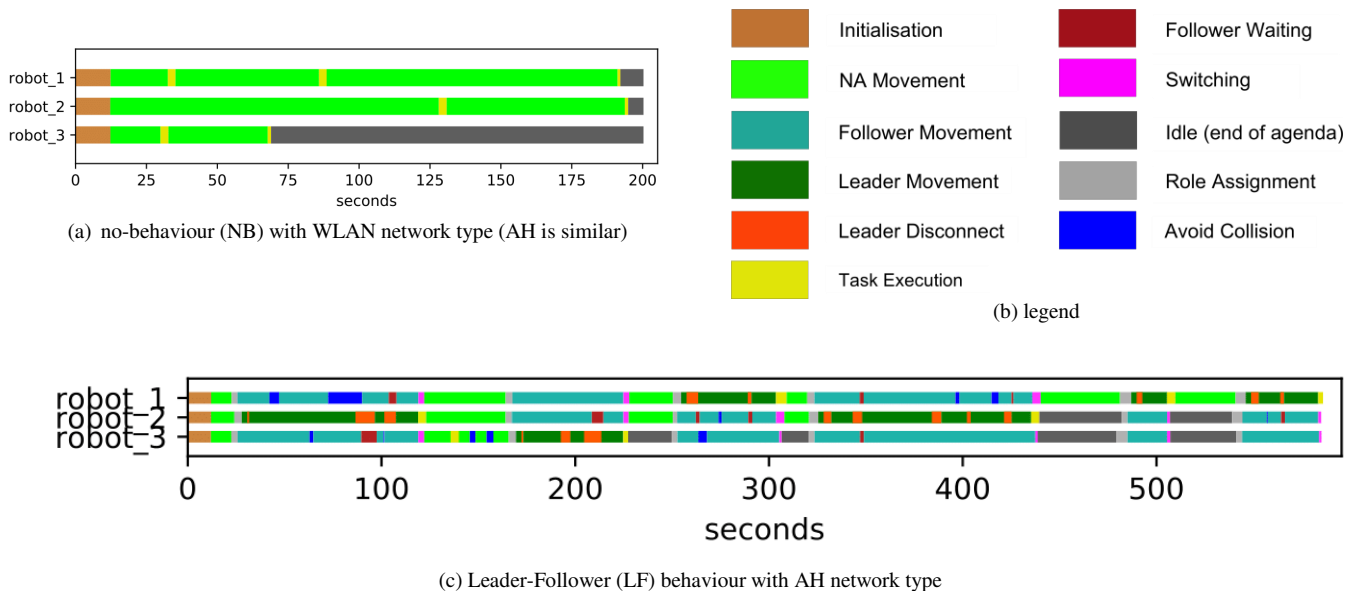


Figure 2. Sample Timelines. These plots illustrate the various activities undertaken by the robots during one representative experiment. Plot (c) shows that the three robots alternate between taking on leader and follower roles.

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. An illustration of role assignment and switching within the LF behaviour is shown in Fig. 2.

IV. EXPERIMENT DESIGN

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. 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 sequentially to each robot R , and the assignments are fixed for all our experiments². To demonstrate the effectiveness of LF in minimising the impact of network connectivity problems, we simulate network perturbation as *simulated packet loss (SPL)*. We compare four values: SPL_j where $j \in \{0, 25, 50, 75\}$ is a percentage of messages that are dropped.

Table I lists the set of experiment configurations. For each, we compared four different network conditions. Each experiment is performed 30 times.

network type	behaviour	network perturbation
WLAN	NB	{ $SPL_0, SPL_{25}, SPL_{50}, SPL_{75}$ }
AH	NB	{ $SPL_0, SPL_{25}, SPL_{50}, SPL_{75}$ }
AH	LF	{ $SPL_0, SPL_{25}, SPL_{50}, SPL_{75}$ }

Table I. Experiment configuration.

We collect a number of different metrics during each experiment. The most relevant metrics discussed here are:

² Robot₁ (the red square in Fig. 3) is assigned tasks $T_1 = \{1, 4, 7\}$, robot₂ (green square) $T_2 = \{2, 5\}$ and robot₃ (blue) $T_3 = \{3, 6\}$.

number of successful tasks, distance travelled and movement time. We expect that the number of successful tasks will decrease when the network is perturbed and connectivity is compromised, except when employing the LF behaviour, which attempts to maintain connectivity. However, we expect that the distance travelled and the amount of time robots spend moving will increase with LF, since they may travel extra distance in order to remain connected.

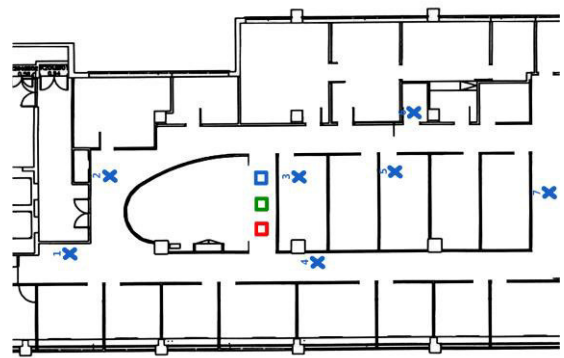


Figure 3. Office setting for experiments, crosses represent task locations and squares robots (based on actual floor plan of building).

V. RESULTS AND DISCUSSION

Fig. 4a shows that LF maintains continuous communication and completes all tasks, whereas NB fails to maintain communication so does not manage to complete all the tasks.

There is a negative impact on using the LF behaviour due to the fact that each robot consistently needs to maintain communication. This leads to the results seen in Fig. 4b, which shows a three-fold increase in distance travelled. However, this is the expected behaviour (i.e., by design) of LF and can be improved in future work. Moreover, Fig.4c shows how LF's movement time is designed differently to that of NB. LF movement time 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.

VI. SUMMARY

We have presented a novel dynamic Leader-Follower behaviour that achieves 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 our framework 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 our dynamic LF behaviour can deal with variable 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. Finally, we hope to explore if other strategies improve the performance of the dynamic behaviour while having less adverse impact on distance travelled and movement time.

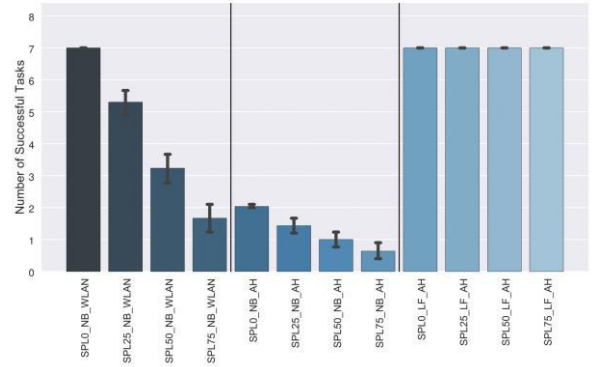
ACKNOWLEDGEMENTS

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

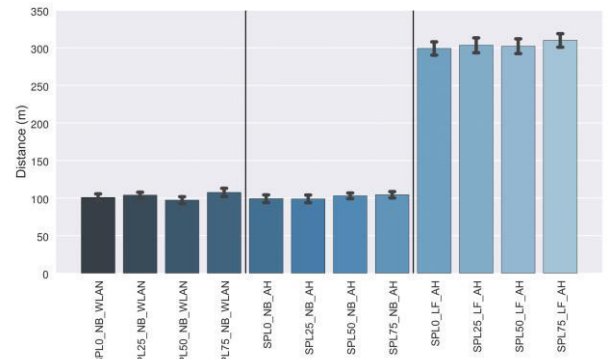
REFERENCES

- [1] T. Zhivkov, E. Schneider, and E. I. Sklar, "Measuring the effects of communication quality on multi-robot team performance," in *Towards Autonomous Robotic Systems: 18th Annual Conference (TAROS)*, Springer, 2017.
- [2] R. Murphy, V. Srinivasan, Z. Henkel, C. Soto, M. Minson, J. C. Straus, S. Hempstead, T. Valdez, and S. Egawa, "Interacting with trapped victims using robots," in *International Conference on Technologies for Homeland Security, HST*, 2013.
- [3] V. Zadorozhny and M. Lewis, "Information fusion for usar operations based on crowdsourcing," in *Proceedings of the 16th International Conference on Information Fusion*, Istanbul, Turkey, 2013.
- [4] M. Lujak, N. Bouraqadi, A. Doniec, L. Fabresse, A. Fleury, A. Karami, and G. Lozenguez, "Towards robots-assisted ambient intelligence," in *5th International Conference on Agreement Technologies*, 2017.
- [5] T. Takahashi, Y. Kitamura, and H. Miwa, "Organizing rescue agents using ad-hoc networks," in *Advances in Intelligent and Soft Computing (AINSC)*, vol. 156, 2012.
- [6] U. Witkowski, M. El-Habbal, S. Herbrechtsmeier, A. Tanoto, J. Penders, L. Alboul, and V. Gazi, "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.
- [7] S. Caccamo, R. Parasuraman, L. Freda, M. Gianni, and P. Ögren, "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.
- [8] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.
- [9] E. Fernandez, T. Foote, W. Woodall, and D. Thomas, "Next-generation ros: Building on dds," *ROSCon*, Chicago, US, 2014.
- [10] A. Dąbrowski, R. Kozik, and M. Maciaś, "Evaluation of ros2 communication layer," *ROSCon*, Seoul, Korea, 2016.

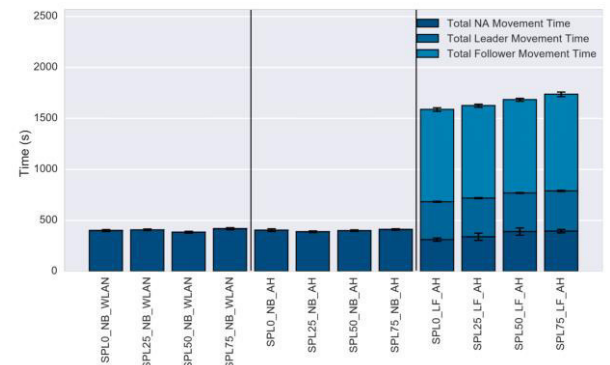
- [11] T. Andre, D. Neuhold, and C. Bettstetter, "Coordinated multi-robot exploration: Out of the box packages for ROS," in *Published in IEEE Globecom Workshops (GC Wkshps)*, Austin, TX, USA, 2014
- [12] E. Schneider, E. I. Sklar, and S. Parsons, "Evaluating multi-robot teamwork in parameterised environments," in *Towards Autonomous Robotic Systems: 17th Annual Conference (TAROS)*, Springer, 2016



(a) number of successfully completed tasks (out of 7)



(b) total distance travelled



(c) total time spent travelling

Figure 4. Experiment Results