2019 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)
Rutgers University, New Brunswick, NJ, USA. August 22-23, 2019

ROS-CBT: Communication Benchmarking Tool for the Robot Operating System

EXTENDED ABSTRACT

Kizito Masaba and Alberto Quattrini Li

I. INTRODUCTION

Simulation plays a critical role in robotics as it allows verification and it provides evaluations of robot software before real world deployment [1]. One prevalent limitation of robotic simulators that support the Robot Operating System, ROS [2] – a de facto standard for developing robotic software – is the absence of communication simulators. This poses a serious challenge in testing software for multi-robot systems (MRS) that use explicit communication, hence limiting the development of this technology. Figure 1 illustrates the problem tackled in this paper.

To address this challenge, we propose the ROS Communication Benchmarking Tool (ROS-CBT). ROS-CBT is a modular, flexible and standalone robotic tool, which simulates communication links between multiple robots in ROS in a *transparent* way – that is, the ROS framework and existing packages do not need to be modified, allowing for easy porting of the robotic software from simulation to real robots. It also provides features for performance evaluation of each communication link at runtime. ROS-CBT is validated with Stage [3] and Gazebo [4], popular robotic simulators integrated with ROS. In general, ROS-CBT can be practically applied for real-time analysis of relevant multi-robot applications, which contributes to accelerate development and test of multi-robot systems.

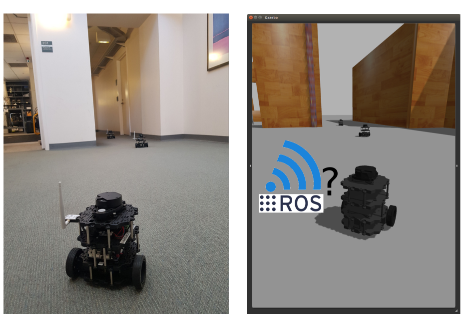

Fig. 1. Given a real multi-robot system equipped with WiFi interface (left) and a corresponding simulation (right) created to perform repeated and a higher number of experiments, the problem we address in this work is: how to simulate a communication network within ROS?

The authors are with Department of Computer Science, Dartmouth College, Hanover, NH USA {kizito.masaba.gr, alberto.quattrini.li}@dartmouth.edu

II. OVERVIEW OF ROS-CBT

ROS is the foundation of ROS-CBT. The fundamental concepts of ROS [2] include:

- *nodes*: processes that perform computation;
- *messages*: typed data structures used for process inter-communication;
- *topics*: named buses over which nodes exchange messages following a publish/subscribe paradigm;
- *services*: communication utility between nodes that is based on Remote Procedure Calls (RPC). It comprises of a named bus and a pair of data structures that define the request and response.

ROS-CBT is integrated within ROS following the above concepts and is designed and implemented with the following characteristics:

- *robotic-simulator independent*: the communication simulator can run along with any robotic simulator;
- *transparency*: the communication simulator should be transparent. Specifically, a roboticist can use the software for a multi-robot system for both simulation and for real robots, without requiring any modification in the code;
- *ease of use*: easy to configure and to run;
- *modularity*: ROS-CBT should permit expansion in communication models and benchmark metrics;
- *low computational cost*: ROS-CBT should have a very low computational footprint not to affect the simulation of other components.

Some existing communication simulation tools simulate communication by synchronizing the robotic simulator (e.g., ARGoS [5]) with a network simulator (e.g., NS2 [6]). An example of such tools is RoboNetSim [7]. This synchronization introduces a delay and computational overhead to the simulator, which are avoided in ROS-CBT: ROS-CBT interacts in a transparent way with the simulator through ROS. In particular, ROS-CBT is a process that runs concurrently with a robotic simulator and intercepts data between robots and delivers them to the recipients according to the communication model – e.g., log-distance path loss [8], [9], Wall Attenuation Factor [10] – by properly remapping the names of topics and services. Figure 2 shows the architecture of ROS-CBT. The communication simulator will keep a data structure for each robot that includes topics and services that share connection with other robots. When a robot produces a message for another robot, the communication

978-1-7281-2876-4/19/$31.00 ©2019 IEEE

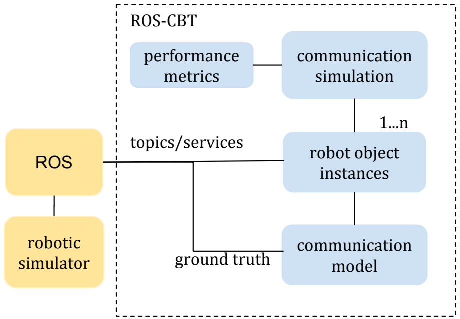

Fig. 2. General structure of ROS-CBT components.

model checks whether the message can or cannot pass using information already exposed by the robotic simulator, such as obstacles and pose of the robots. Note that there can be multiple communication models, depending on the simulated communication interface. In our implementation, we provide four communication models: unlimited communication, disk, log-distance, and wall attenuation factor (WAF). Furthermore, ROS-CBT implements performance metrics such as throughput, bandwidth, and signal strength for analyzing and benchmarking coordination algorithms.

III. EXPERIMENTS AND DISCUSSION

We test ROS-CBT, currently implemented in Python, on two representative multi-robot scenarios: indoor mapping and outdoor coverage. Both scenarios demonstrate the broad applicability of ROS-CBT for testing multi-robot systems on different simulators and ROS packages. Due to space limitation, we only discuss the indoor mapping scenario.

The indoor mapping case involves the use of Stage 2D robotic simulator [3] with a team of 6 Pioneer 2AT robots, whose task is to build individual maps of the environment, following a frontier-based strategy [11]. Poses and scans are shared with other robots to integrate all measurements in a map using a graph-based Simultaneous Localization and Mapping (SLAM). In addition to the existing performance metrics, we implement a task independent metric, *explored area*, measured as a percentage of explored area, in order to best analyze the performance of this task.

For example, Figure 4 shows the distance between robot 6 and the other robots, signal strength, and bandwidth of an

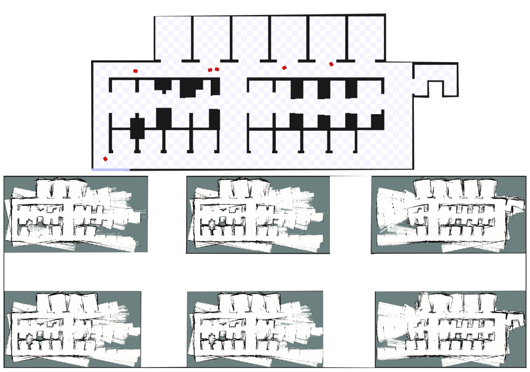

Fig. 3. Snapshot of an ongoing simulation of indoor mapping experiment with six robots (top) and the corresponding generated maps for each robot (bottom).

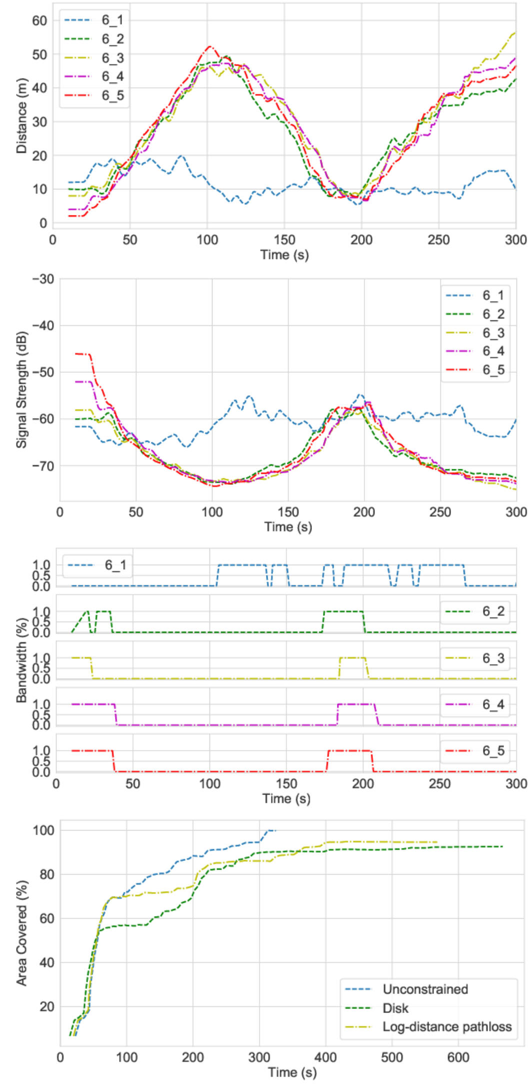

Fig. 4. Analysis of the indoor mapping simulation with 6 robots, when running ROS-CBT with the log-distance path loss model. First: distance between robots i and j. Second: signal strength between robots i and j. Third: bandwidth of the communication link, i-j. Fourth: percentage of explored area by robot 4 over time under different communication models.

experiments with 6 robots (see Figure 3) under log-distance path loss communication model. This allows researchers to analyze the impact of a communication model to the indoor mapping task as shown in the last plot of Figure 4 and develop new communication-aware exploration strategies – e.g., a strategy that takes into account episodic rendezvous might improve the performance of the multi-robot system [12].

Overall, ROS-CBT proposed design and implementation allow for modular communication models and transparent integration within the existing ROS framework without requiring any modification of existing ROS packages. This is a powerful tool for testing multi-robot system software before real-world deployment. The code is opensource and will be available at [13]. Future work includes implementing and validating other communication models and simulating all the layers of the communication stack.

REFERENCES

[1] F. Amigoni and V. Schiaffonati, “Good experimental methodologies and simulation in autonomous mobile robotics,” in *Model-based reasoning in science and technology*. Springer, 2010, pp. 315–332.

[2] M. Quigley, K. Conley, B. 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*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[3] R. Vaughan, “Massively multi-robot simulation in Stage,” *Swarm intelligence*, vol. 2, no. 2-4, pp. 189–208, 2008.

[4] N. Koenig and A. Howard, “Design and use paradigms for gazebo, an open-source multi-robot simulator,” in *Proc. IROS*, vol. 3. IEEE, 2004, pp. 2149–2154.

[5] C. Pinciroli, V. Trianni, R. O’Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. Di Caro, F. Ducatelle, *et al.*, “ARGoS: a modular, multi-engine simulator for heterogeneous swarm robotics,” in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 5027–5034.

[6] “ns2,” https://www.isi.edu/nsnam/ns/, 2019.

[7] 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.

[8] T. S. Rappaport, *Wireless communications: principles and practice*. Prentice Hall PTR New Jersey, 1996.

[9] A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005.

[10] P. Bahl and V. N. Padmanabhan, “RADAR: an in-building RF-based user location and tracking system,” in *Proc. INFOCOM*, vol. 2, 2000, pp. 775–784.

[11] B. Yamauchi, “Frontier-based exploration using multiple robots,” in *Proc. International Conference on Autonomous Agents*, 1998, pp. 47–53.

[12] V. Spirin, S. Cameron, and J. de Hoog, “Time preference for information in multi-agent exploration with limited communication,” in *Proc. TAROS*, 2013, pp. 34–45. [Online]. Available: https://doi.org/10.1007/978-3-662-43645-5_5

[13] K. Masaba, “ROS Communication Benchmarking Tool,” https://github.com/dartmouthrobotics/roscbt-generator.git.