Utilizing Awareness of Surroundings to Reduce Network Disruption While Maximizing Coverage in Multi-Robot Exploration #2221

Khan Henderson, Khang Vo Huynh, Ethan Kilmer
Mentored by Prof. Elizabeth Jensen

ST. OLAF COLLEGE

Abstract

Multi-robot exploration encompasses a variety of applications including deep sea, space, and search and rescue. In many of these, maintaining communication between robots can be crucial. However, communication networks can experience interference and partitioning from external networks or physical objects. Many proactive algorithms focus on the distance between the robots, but proximity to physical objects can have notable and unexpected effects as well. We collected data on the communication strength and reliability between neighboring agents and the distances to nearby objects in multiple indoor and outdoor scenarios. Utilizing machine learning models, the robots will be capable of reacting to their current situation as they attempt to maximize the exploration coverage while minimizing disruptions to communication.

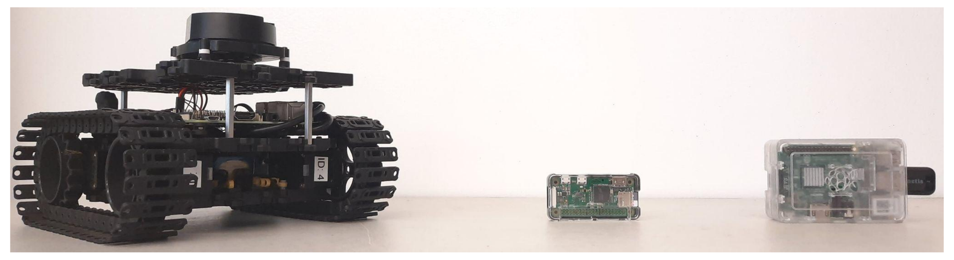
Goals

1. Set up an ad hoc network and integrate its use with ROS 2.
2. Data Collection:
   a. Measure signal strength between agents in a variety of scenarios, paying close attention to impact of physical objects..
   b. Identify situations in which the network connection is dropping due to physical objects not just distance.
3. Future: Create models to use in determining robot movement.

Elements of Our Multi-Robot System

Hardware Devices

- Each device is an ***agent*** in the system.
- Mobile Turtlebot3 robots
  - Modified to use tank treads to move across complex terrain.
  - OpenCR board provides access to motors and sensors.
    - Challenge: Changing the firmware to control additional motors.
  - Raspberry Pi 3 to develop and launch programs.
- Stationary Raspberry Pi Zero Ws
  - Challenge: When outside, the agents would unexpectedly shutdown due to heat and wind.
- Two Raspberry Pi 3s that provide external access to the system.



Left to right: a Turtlebot3, a Pi Zero W, and a Pi 3.

Robot Operating System (ROS) 2

- ROS 2 is middleware for developing robot programs. It provides:
  - A distributed system that allows for hardware independence, meaning that any robot could implement it.
  - Access to motor and sensor data through publisher/subscriber topics or service requests.
    - Example: The LiDAR publishes scan data to a topic, which a program subscribes to for access.
- Our program collects proximity data from the LiDAR and network data using system commands. The components are diagrammed below.

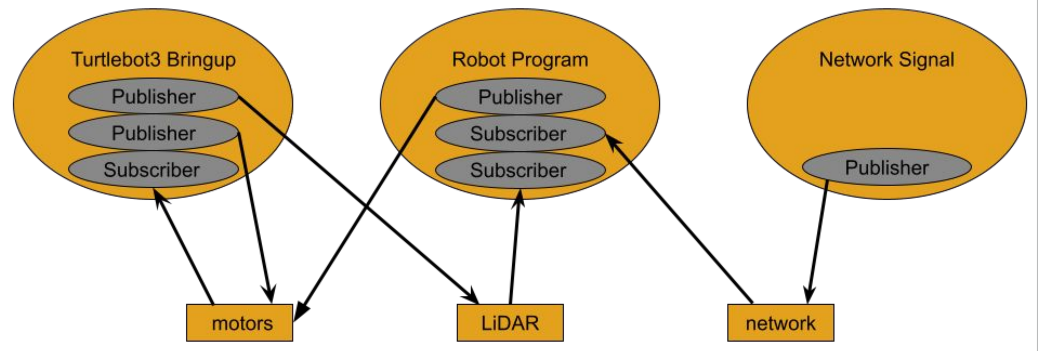
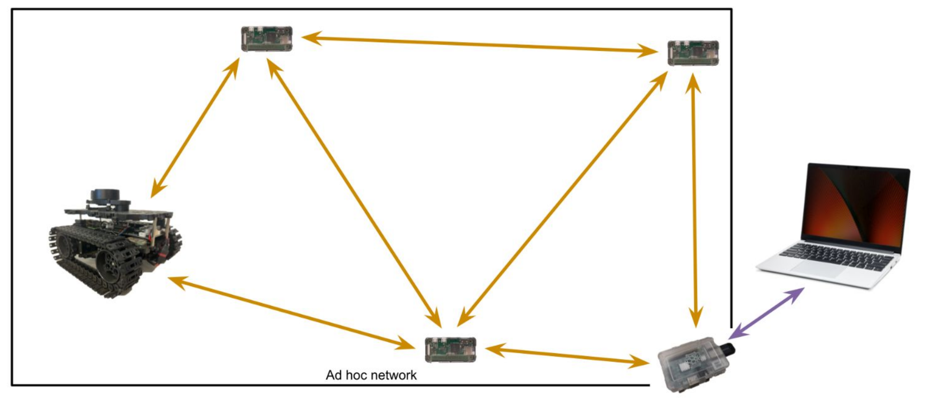


Diagram of ROS 2 programs, topics, and interactions.

Networking

- Agents communicate over a mobile ad hoc network (MANET) using the B.A.T.M.A.N. IV routing algorithm.
  - An ad hoc network allows agents to join and leave the network at will.
  - Challenge: Automating setup of an ad-hoc mesh network on 40 nodes.
- In networking, each communicating agent is referred to as a ***node.***
  - Nodes communicate with each other directly when they are in range and through multi-hop routing otherwise.
  - The algorithm keeps the routing tables updated by regularly collecting neighbor transmit quality.



Network set-up with multiple types of nodes.

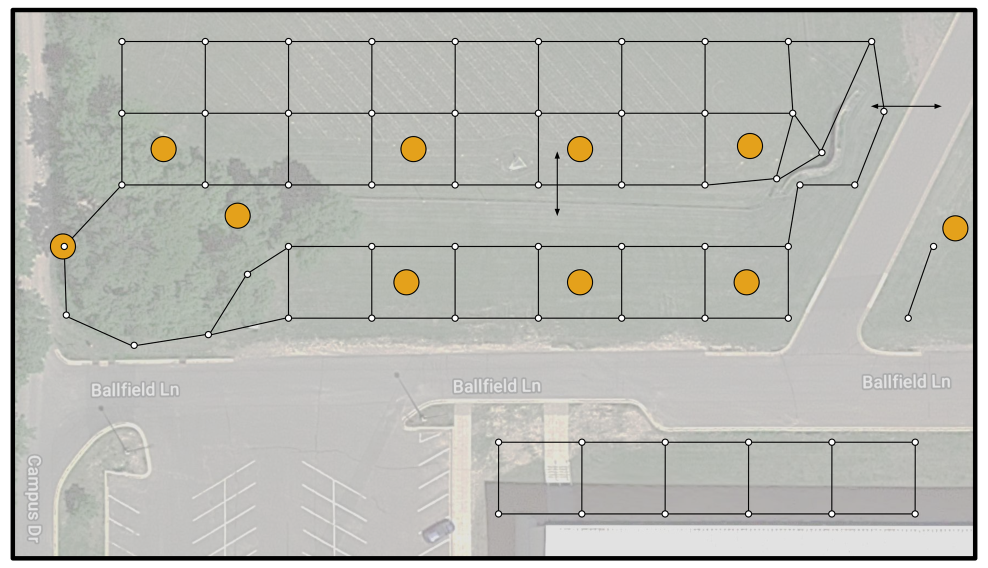
Data Collection

We created multiple scenarios in which we placed Raspberry Pi Zero Ws in the environment such that each had at least one connection at all times, no matter where the robot or access point were located. Then we moved the robot along a specific path stopping at regular intervals to collect proximity and direct neighbor signal data. We collected data both inside and outside. In both cases, we also ensured that the environments included open spaces with little to no physical obstacles as well cluttered spaces with large physical obstacles such as walls and barriers.

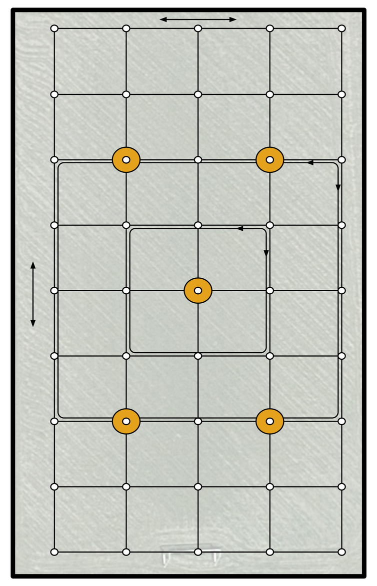
Scenario #1: Indoor, hallway and open area.

The three scenarios depicted to the left and below were used to generate a variety of samples for use in our later modeling steps.

Legend for scenarios:
- stationary Pi Zero Ws
- data collection location
- robot movement direction



Scenario #2: Outdoor, trees, retaining walls, and open areas.



Scenario #3: Outdoor, open.

Due to the high volume of data samples collected, we will display our results on screen. You may also access the results yourself using the following QR code:

Conclusion

Physical distance combined with physical obstacles, such as walls, can cause the robot to lose connection. TQ data appeared to reliably indicate an impending loss of connection, being a responsive monitor of signal strength. RTT data reflected network traffic in the area, but collecting RTT values became sluggish particularly when signal strength was low, which may be problematic for a behavior model making live decisions. Our findings could help diagnose the state of the communication network in a multi-robot system, and highlights the unpredictability of these networks. Our future plans for fall semester include developing machine learning models using decision trees and neural networks. We will then load these models on the robot so it can predict network connectivity and performance based on its surroundings.

Acknowledgements

We are grateful to the CURI operating budget and St. Olaf Endowment for Undergraduate Research in Mathematics that funded this project.