foxySwarm Project Plan

A package that provides ad-hoc communication in a swarm

M. Raşit Özdemir

KOVAN RESEARCH LABORATORY, MIDDLE EAST TECHNICAL UNIVERSITY

GITHUB.COM/ROBOFOXY/FOXYSWARM

This project was done under the supervision of Dr. Erol Şahin with the infrastructural support of KOVAN Research Laboratory.

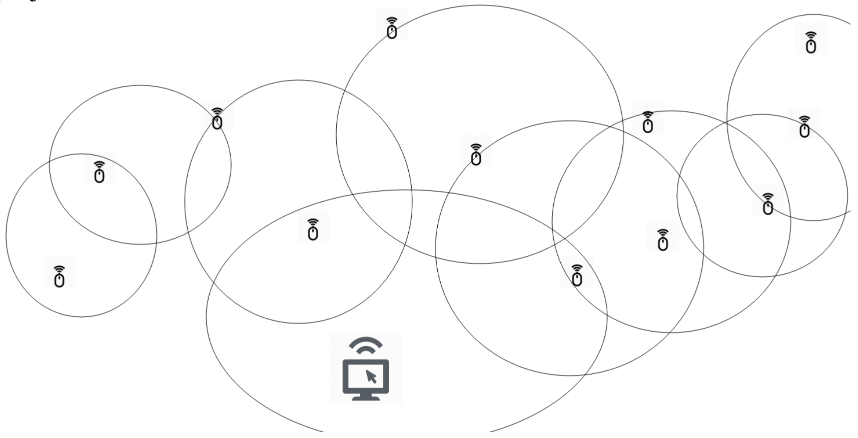
First version, October 2018

Contents

1. Plan / Requirements

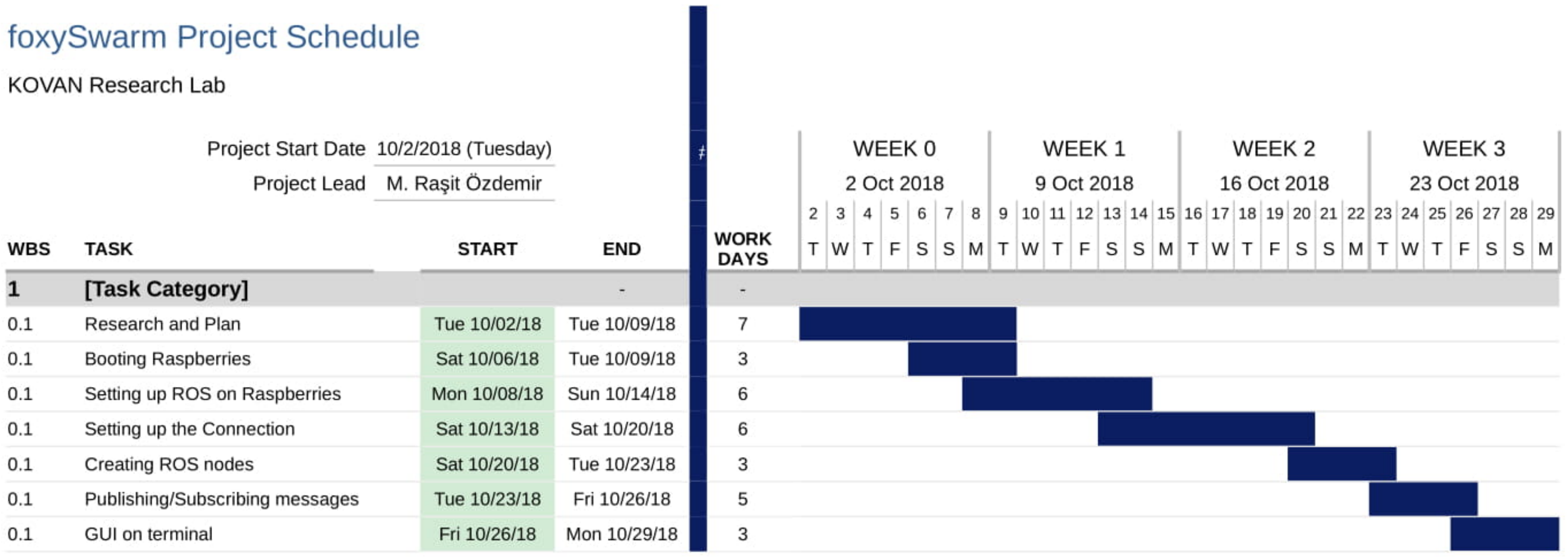
1.1 Objective

This document aims to represent the project plan of foxySwarm. The end product for this project is a ad-hoc communication application between Raspberries and a PC. There will be Raspberry Pi B+ devices, and a PC. A suitable ROS compatible operating system should be installed on the single board computers. Then, a communication protocol will be defined and implemented at the end of the project.



1.2 Steps

- Do research to specify the requirements.
- Boot Raspberry Pi 3 B+.
- Install Robot Operating System on Raspberries.
- Set up network connection between Raspberry-Raspberry and PC-Raspberry.
- Create proper ROS nodes.
- Configure publishing/subscribing routines.
- Create a graphical user interface (GUI).



foxySwarm Project Schedule

KOVAN Research Lab

Project Start Date 10/2/2018 (Tuesday)

Project Lead M. Raşit Özdemir

WBS	TASK	START	END	WORK DAYS
1	**[Task Category]**		-	-
0.1	Research and Plan	Tue 10/02/18	Tue 10/09/18	7
0.1	Booting Raspberries	Sat 10/06/18	Tue 10/09/18	3
0.1	Setting up ROS on Raspberries	Mon 10/08/18	Sun 10/14/18	6
0.1	Setting up the Connection	Sat 10/13/18	Sat 10/20/18	6
0.1	Creating ROS nodes	Sat 10/20/18	Tue 10/23/18	3
0.1	Publishing/Subscribing messages	Tue 10/23/18	Fri 10/26/18	5
0.1	GUI on terminal	Fri 10/26/18	Mon 10/29/18	3

1.3 Initial Plan

Raspberri Pi 3 B+ is the latest model of the Raspberries. Therefore, there are a lot of incompatibility between Ubuntu and ROS versions. Firstly, I created a github repository and tried to boot Raspberries with Raspbian operating system.

1.3.1 Initial Problems

- Raspbian is the most suitable operating system for the single board computers we have. However, it has some problems with ROS.
- There is no announced Ubuntu versions that is compatible with Raspberry Pi 3 B+.
- There is not any external monitor and HDMI cable in my home.

1.3.2 Solutions to Initial Problems

- All Raspbian versions are eliminated because of ROS requirements.
- After a long research and surfing the Internet, I found a way to hack Ubuntu 16.04 compatible with Raspberry Pi 3 B+.
- I already needed an external monitor in my home. Then, I bought one.
- Burak Hocaoğlu provided a HDMI cable temporarily.

1.4 Implementation Part

1.4.1 Network Design Options

Since, we want to create an ad-hoc network between anonymous air vehicles, we can not use promiscuous networking mode for wireless interface. I did a research about networking mode that we should use in this project and found two options. Monitoring mode and ad-hoc mode with common SSID's.

Monitor mode allows a computer with a wireless network interface controller to monitor all traffic received from the wireless network. But unlike the more commonly used promiscuous mode, monitor mode allows packets to be captured without having to associate with an access point or ad-hoc network first. In this mode, we are able to broadcast IEEE 802.11 raw packets. However, we'll need to add a second USB WiFi dongle to our Raspberry Pi. Because BCM43438 WiFi chipset used on Raspberry Pi3 does not support monitoring mode. Although it will be a cheap hardware requirement, it is more expensive than the free one.

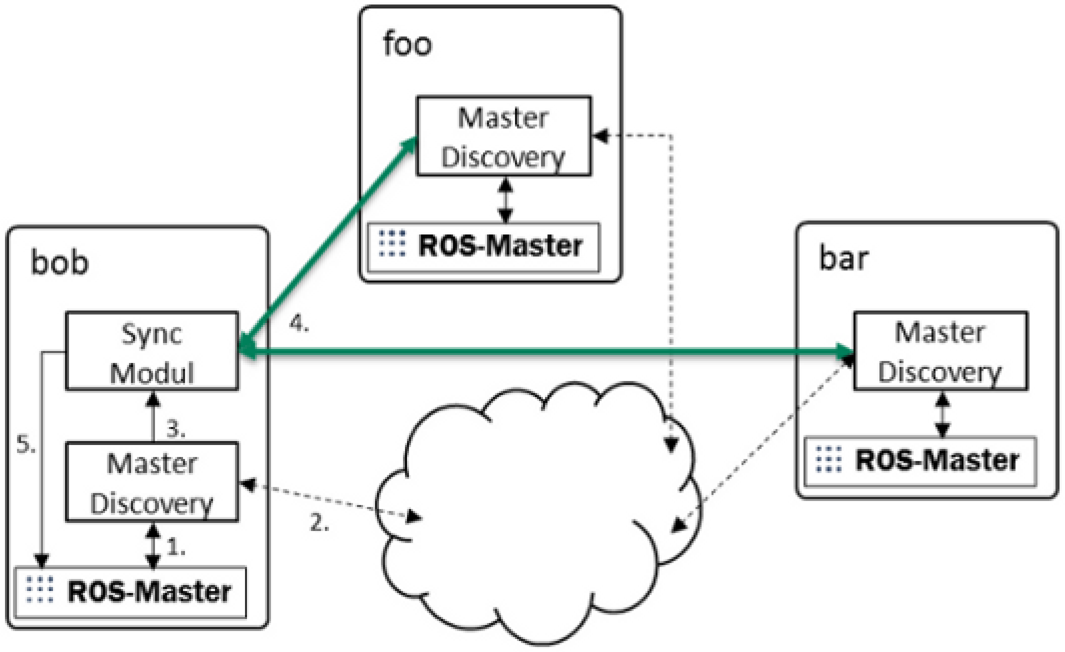
A wireless **ad hoc network** or MANET (Mobile ad hoc network) is a decentralised type of wireless network. By changing the settings of wireless interface and rebooting, we can set up a ad-hoc network. However, in ad-hoc mode, we cannot connect other wireless networks, but we can open a socket and publish/listen messages from other Raspberri Pi's that has the same network configuration with same SSID.

We can easily configure the wireless interface mode by selecting a static IP by editing /etc/network/interfaces. Since there will be multiple drones in the swarm, they should not choose the same IP.

After some testing(creating socket and binding it from other device) with ad-hoc mode, we decided to use ad-hoc mode.

1.4.2 Implementation

The ROS package **multimaster_fkie** provides a set of ROS nodes to establish and manage a multimaster network. The **master_discovery** node connects to the ROS-Master, gets changes by polling and publish the changes (multicast or/and unicast) over the network. The received changes of the remote ROS-Master are published to the local ROS topics. The **master_sync** node connects to the discovered master_discovery nodes, requests the actual ROS state and registers the remote topics/service.



The 'master_discovery' node sends periodically a multicast message to notify about an available ROS master. After discovering the ROS-masters, 'master_sync' node synchronizes the local ROS master to remote ROS masters discovered by master_discovery node.

A script is implemented to set up the ad-hoc network configurations. Moreover, installation of multimaster_fkie package is done by execution of this script.

poseBroadcaster

poseBroadcaster node publishes fake geometry_msgs/PoseStamped messages to poseBroadcaster topic. All the drones that can notify the local ROS-master can subscribe the published messages.

poseSubscriber

poseSubscriber node subscribes the messages that are published into poseBroadcaster topic from discovered ROS-masters. In this node, library curses is used to monitor the observable drone poses to the user.

TrajectoryPolynomialPiece

TrajectoryPolynomialPiece is a custom message that is defined in kovanSwarmUAV project. destination entity is added to specify the ROS_HOSTNAME of destination drone in the swarm.

```
string destination
float32[] poly_x
float32[] poly_y
float32[] poly_z
float32[] poly_yaw
duration duration
```

trajectory_publisher

trajectory_publisher is a node that publishes different fake trajectories to all the drones in the network by specifying destination entity of TrajectoryPolynomialPiece message type. First of all, it should subscribe /master_discovery/linkstats topic to learn the ROS_HOSTNAMEs of the drones in the network. Then, it publishes a fake trajectory to trajectoryPublisher topic. In order to simulate well, the sizes of arrays are 1000 for all pose directions.

drone_core

drone_core node is implemented to develop all the drone core functionality inside. For now, it can subscribe trajectoryPublisher topic to get the trajectories that are published to it.

1.5 Demo

There will be a demonstration session on second week of November, 2018 at Dr. Sahin's office. Hop-by-hop forwarding of messages will be completed until end of that week.

This document is to be edited during the development.