Team G- Robographers

Progress Review 8

Individual Lab Report 7

Jimit Gandhi

Team Members

Rohit Dashrathi

Sida Wang

Gauri Gandhi

Tiffany May

1.0 Introduction

This week the team focused on setting up a multi-master turtlebot environment and got familiarized with Gazebo simulator

2.0 **Individual Progress and Contribution**.

The goals for our Progress Review 8 were as follows

1. Install Ubuntu 14.04 on the new chrome-book Acer C720 and install ROS Indigo.
2. Install Rocon multi-master package.
3. Setup the Rocon bringup for Turtlebots.
4. Get familiarized with Gazebo.
5. Implement teleoperation of multiple robots on gazebo which communicate with each other using the multi-master Rocon framework

Last year, Allard's team experimented implementing a swarm controller using a single master framework for multiple Turtlebot nodes and try controlling the Turtlebot using tele-operation. Now for tele-operation or for moving a Turtlebot from one place to another, we need to publish the velocity values on the topic cmd_vel_mux. The team last year tried out this approach. But the topics seem to interfere with each other and all the Turtlebots receive the velocity commands even though it was not intended. Thus multi-master set up was required for implementing a robust swarm controller. ROS provides Rocon package which is called Robotics in Concert.

My major work was understanding how Rocon framework works. The framework is similar to the LAN networks that we have at our homes or offices. There is a central server called the 'hub'. Structurally hub is a form of local ReDiS (Remote Dictionary Server) similar to the one used in LAN connections. The purpose of this hub is to collect all the data coming from the different nodes and direct them to the desired nodes thus preventing high traffic by not sending the messages to all the nodes and also keeps all the nodes decoupled thus regulating the message passing. Between the connection of node's master and the hub there is a 'gateway' which is similar to a public interface for the hub to interact with. Whenever a message is received by the hub, the hub checks something called gateway's 'switch'. If this switch is flipped by the user or the node the hub passes on the message to this node. If the switch is not flipped then the message is not sent to the node. The gateways of other nodes can also interact with the gateways of other nodes. If a node wants a service from a particular node, the gateway of the former will request a ‘pull’ from the gateway of the latter. So this is an overview of how the Rocon framework works. The figure 1 below is an illustration

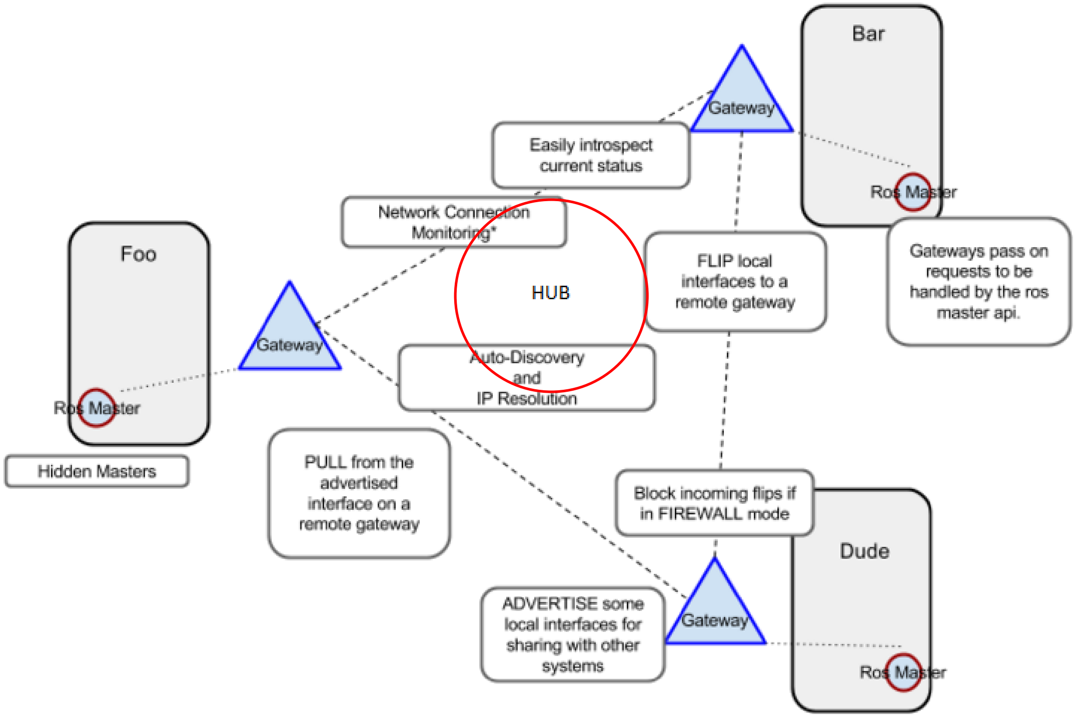
Bar
Gateway
Easily introspect current status
Ros Master
Network Connection Monitoring*
Foo
FLIP local interfaces to a remote gateway
Gateways pass on requests to be handled by the ros master api.
HUB
Gateway
Auto-Discovery and IP Resolution
Ros Master
Hidden Masters
PULL from the advertised interface on a remote gateway
Block incoming flips if in FIREWALL mode
Dude
ADVERTISE some local interfaces for sharing with other systems
Gateway
Ros Master

Figure 1: Rocon Gateway Model with three ROS Masters. The Red circle represents HUB which monitors the connectivity.

After understanding the way these networks work, I installed the Rocon package along with its dependencies on few other libraries. Since for Rocon we need two or more laptops to serve as a host and a client, I Gauri and Sida decided to work together. After installing, all three of us explored the tutorials of the Rocon framework. The Rocon tutorials are quite vast since it covers many different situations. Few examples are multiple masters on single computer tutorial, Hub launch tutorial, Gateway tutorial, Flipping tutorial, Concert tutorial. I covered most of the relevant tutorials and learnt how to write gateway files and flip the topics for any node manually. Apart from this there were some ROS’s usual turtle sim examples which were worth taking a look at. The concert files of turtle sim helped me how to write a concert file for multi-master setup. A concert file is a collection of more than one launch files as well as launch of hub. One has to mention the port number – an important parameter that assigns a channel of communication of a particular node to the hub. By default the hub is given the port as number 11311. Rest of the nodes can be configured as desired. So in one of the concert files I made, I assigned the two nodes other nodes as 11312 and 11313 respectively.

After learning about the Rocon package, I Gauri and Sida tried to write multi-master concert files on which one node would constantly publish “Hello ROS!!” and two other nodes would constantly listen to this topic. We even tried certain experiments with changing ZeroConfig settings and also manually turning the flip off and on the topics to the listener nodes. The ZeroConfig setting enables the hub to discover locally available devices on the same network. One has to enable the ZeroConfig in order to discover these devices otherwise the Hub will not detect them automatically and thus will never connect with them. In order to monitor the node conditions and connectivity, I also learnt how to use the Rocon package interface called RoconApp Remocon which is a real time interface where one can see the connectivity maps. Figure 2 shows how multiple nodes and masters connect to each other.

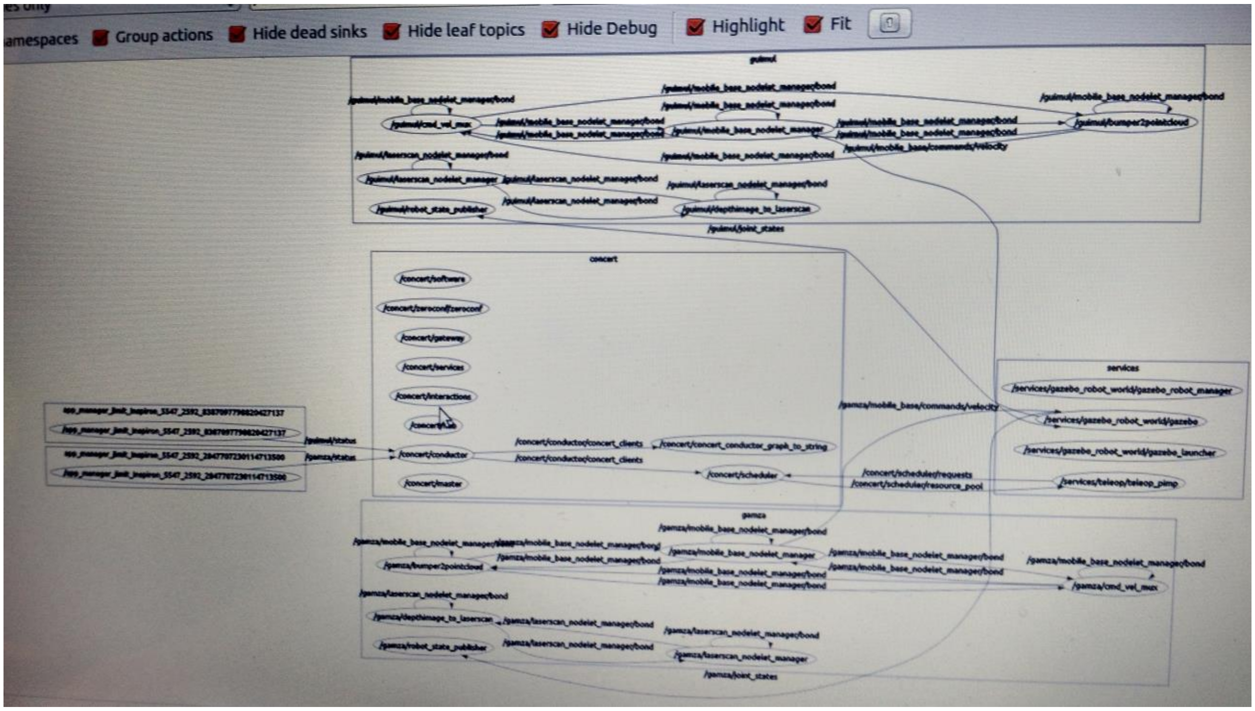
amespaces
Group actions
Hide dead sinks
Hide leaf topics
Hide Debug
Highlight
Fit
guimul
/guimul/mobile_base_nodelet_manager/bond
/guimul/cmd_vel_mux
/guimul/mobile_base_nodelet_manager
/guimul/bumper2pointcloud
/guimul/mobile_base/commands/velocity
/guimul/laserscan_nodelet_manager/bond
/guimul/laserscan_nodelet_manager
/guimul/robot_state_publisher
/guimul/depthimage_to_laserscan
/guimul/joint_states
concert
/concert/software
/concert/zeroconf/zeroconf
/concert/gateway
/concert/services
/concert/interactions
/concert/hub
/concert/conductor
/concert/master
/concert/conductor/concert_clients
/concert/concert_conductor_graph_to_string
/concert/scheduler
/concert/scheduler/requests
/concert/scheduler/resource_pool
app_manager_jimit_Inspiron_5547_2592_8387097798820427137
/app_manager_jimit_Inspiron_5547_2592_8387097798820427137
app_manager_jimit_Inspiron_5547_2592_2847707230114713500
/app_manager_jimit_Inspiron_5547_2592_2847707230114713500
/guimul/status
/gamza/status
services
/services/gazebo_robot_world/gazebo_robot_manager
/services/gazebo_robot_world/gazebo
/services/gazebo_robot_world/gazebo_launcher
/services/teleop/teleop_pimp
/gamza/mobile_base/commands/velocity
gamza
/gamza/mobile_base_nodelet_manager/bond
/gamza/mobile_base_nodelet_manager
/gamza/bumper2pointcloud
/gamza/cmd_vel_mux
/gamza/laserscan_nodelet_manager/bond
/gamza/depthimage_to_laserscan
/gamza/robot_state_publisher
/gamza/laserscan_nodelet_manager
/gamza/joint_states

Figure 2: Each box is a separate node. The box in the center represents hub. The one above and below the hub are the simulated Turtlebots on Gazebo and bot are connected to the Teleoperation node one the right box.

Rohit and Tiffany were working on Gazebo part. They learnt the Gazebo tutorials and gave me a quick tutorial of these tutorials. Rohit showed me how to initialize a single Turtlebot teleoperation node on Gazebo simulation and how to control it. After playing around with it and investigating the launch files for a single simulated Turtlebot node, I created a concert file for launching multiple master Turtlebot nodes which I learnt during the Rocon tutorials. I configured the settings of the port as well as those of the gateways. A few trials and debugging exercises later, I could successfully tele-operate two different Turtlebots on Gazebo. I could manually select which Turtlebot I wish to move. I also launched three Turtlebots, although it was successful at times, sometimes one or two Turtlebots were unable to be tele-operated. The reason for this was that the gateway IP would sometimes fail to connect with the hub. Figure 3 shows teleoperation and figure 4 show the Turtle bot simulation on Gazebo

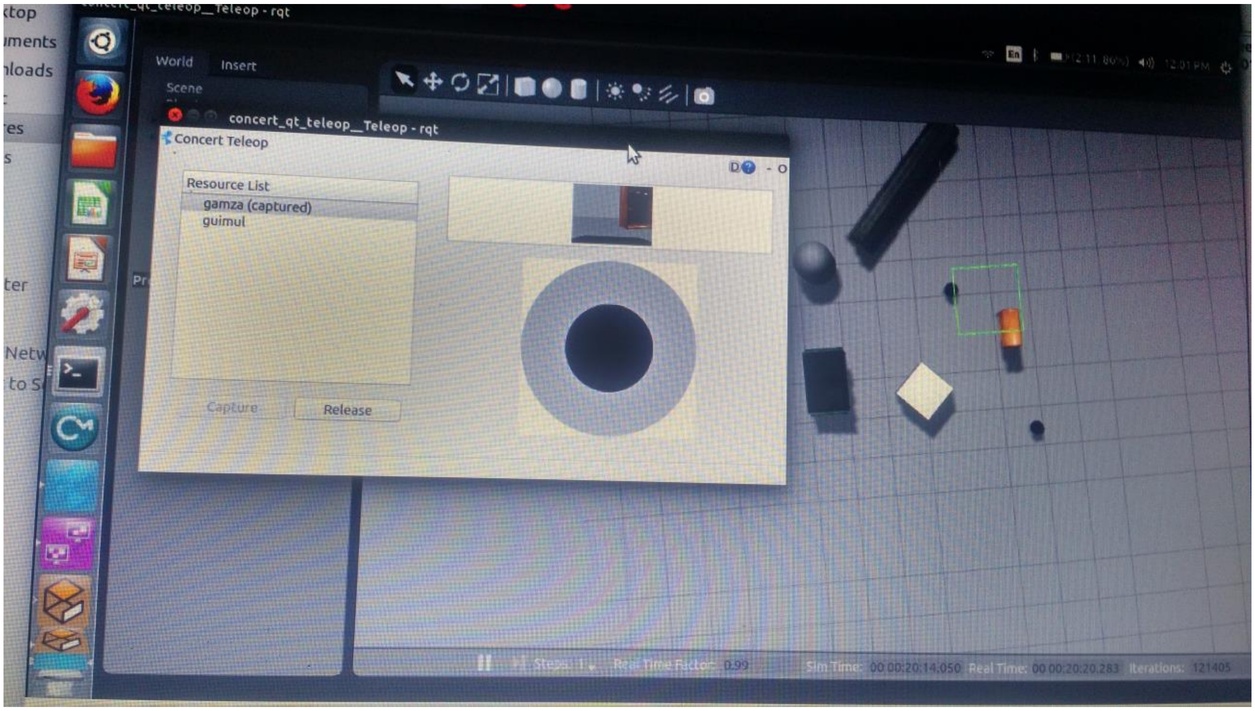

Figure 3 : Teleoperation of two Turtlebots on Gazebo. The names of Turtlebots are gamza and Guimul.

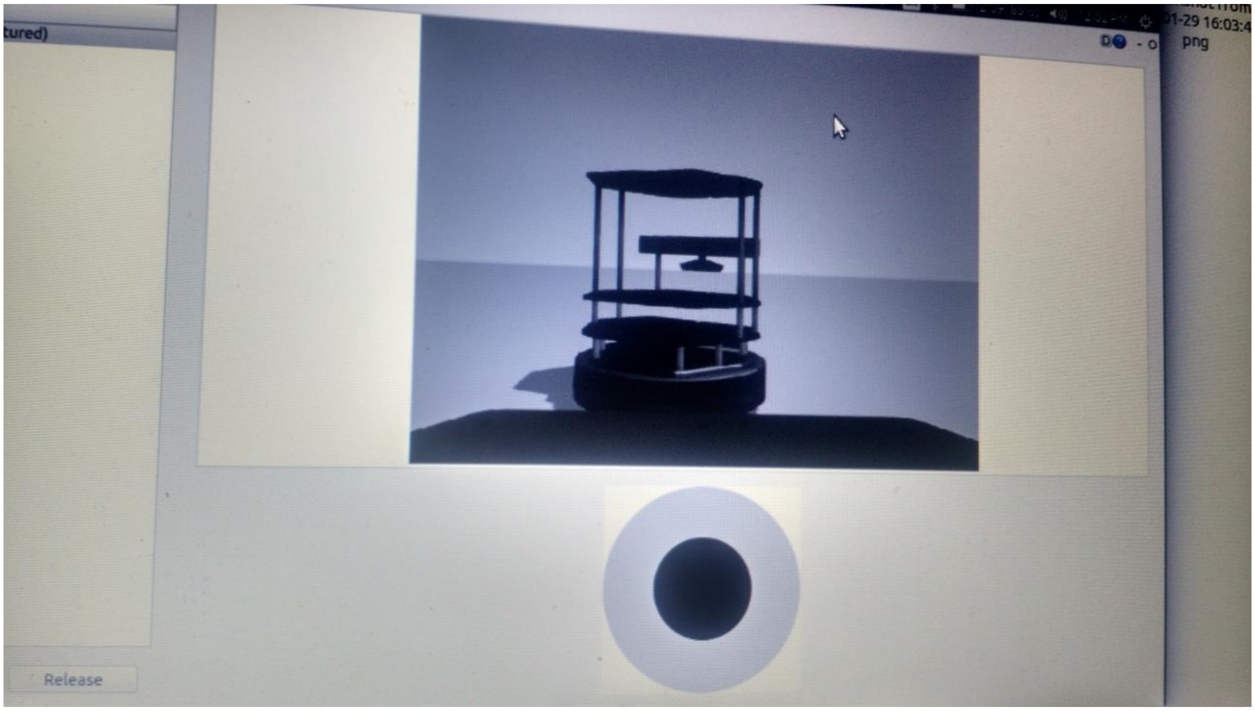

Figure 4: Turtle bot model on Gazebo.

2.1 Challenges Faced

The challenges that I faced was while setting up Rocon. There were many settings to be configured. I was not aware of the importance of flipping of the node topics initially. Hence even if my nodes would launch successfully, they would never receive the published messages. I was stuck on this for a long time as there were not many questions or FAQs on Rocon since this framework is only used by the niche roboticists involved with swarm research. However, I read few ILRs of Allard Dupius, who was responsible for setting up Rocon in last year's team project. He had faced similar issue about flipping and his ILR gave me a direction in which to proceed. However, after hours of due diligence, I was successfully able to get the topics published and subscribed at my will.

Another problem that I faced was again my ignorance about ZeroConfig settings and Daemon Avahi settings. These two settings are responsible for discovering and connecting with the nearby devices on the local network. One has to manually enable ZeroConfig and disable Daemon Avahi setting in order to get the nodes connect successfully. Once setting them correctly I also discovered that one has to reboot the OS in order to put them in effect. This realization also came to me after spending quite a while going round and round trying almost every combination of these settings minimum 3 times.

A minor challenge was faced by me Gauri and Sida while we were trying to set up a multi master launch on multiple machines. We have to set up a Master_Url as the server's or the hub's IP while also defining the port number as 11311. We learnt this after few internet searches and also going through Allard's ILR and repository. We could not directly use Allard's code since he implemented all the packages on ROS Hydro Ubuntu 12.04 while we are using ROS Indigo on Ubuntu 14.04. But we did find many answers in Allard's ILR.

Sida had installed Ubuntu 14.04 on Chromebook successfully with ease. The major challenge came after when I tried to restart it. Due to my unfamiliarity of chrome OS and since Sida was not present at that time, I accidentally turned off something called 'Developer's mode' of the Chrome OS. Unaware of the consequences, I searched on internet on how to switch back to Ubuntu as I was under the impression that this is a dual boot system just like Windows and Ubuntu. After trying all the solution that I could find out there, I failed to switch from ChromeOS to Ubuntu. This went for 3-4 hours, I was constantly trying when I read somewhere that if one turns off the Developer mode then it erases all local data as well as any other OS mirrored on the ChromeOS. I and Sida finally recovered the entire system and reinstalled it again the next day. It was not a very difficult challenge but it was time consuming and exhaustive one.

3.0 Team Work

Rohit and Tiffany worked on learning Gazebo from scratch following tutorials. Rohit performed tele-operation of simulated Turtlebot on the Gazebo while Tiffany created a world model of the room or the final venue of our SVE with three Turtlebots moving within the room at constant velocity. Sida worked with Gauri mostly on learning Rocon as well as setting up config files for multi-master launch on multiple machines. Both of them taught me practical things about using Rocon such as setting master_url and enabling the ZeroConfig. Sida also installed Ubuntu 14.04 twice on chrome book and then installing ROS and other dependencies.

3.1 Future Work

We plan to move as per our work breakdown structure. Next week's agenda includes the following.

1. Implement swarm by making robots move from one location to another using Flocking algorithm on Gazebo until new chrome-books arrive.
2. Implement multi-master launch of Intraface and enable collaborative facial expression and photo-clicking.
3. Since the lab is using their network they asked us to order a new router and set up a new network in order to have a private Rocon network for Robographers.

4.0 References

1. Intraface Software - Fernando De la Torre† , Wen-Sheng Chu† , Xuehan Xiong† , Francisco Vicente† , Xiaoyu Ding† , Jeffrey Cohn†‡ †Robotics Institute, Carnegie Mellon University, (http://www.humansensing.cs.cmu.edu/intraface/)
2. A gentle Introduction to ROS https://cse.sc.edu/~jokane/agitr/
3. www.stack-over-flow.org
4. http://wiki.ros.org/rocon/indigo/Guide
5. http://wiki.ros.org/gazebo
6. https://sites.google.com/site/mrsdproject201415teamc/
7. www.ros.org
8. www.opencv-tutorials.org