

Progress review #10

Gauri Gandhi

Team G Robographers

Teammates:

Rohit Dashrathi

Jimit Gandhi

Tiffany May

Sida Wang

ILR #9

March 17, 2016

a. Individual Progress

For the tenth progress review, I worked with Jimit to develop a multi-master system for detecting facial expressions and capturing the best photo using ROCON and Intraface.

I. Creating Rapps

ROS Indigo has introduced the concept of Rapps for setting the multi-master system using ROCON. Rapps are meta packages providing configuration, launching rules and install dependencies. As a first step, I created a dedicated package for Rapps on each of the Chromebook. After that I created an emotion_talker and an emotion_listener node that published and subscribed respectively on the emotion_chatter topic. The published message was the confidence value of the happiness emotion published by the Intraface node. The message type was std_msgs/Float32. Then, in the Rapps package, I created the following files:

- (i) Photo_click.launch file to launch the emotion_talker node and the modified photo_click node created by Jimit
- (ii) Photo_click.interface file to list the publisher and subscriber topics to be shared over the ROCON. For our case, the publisher was the emotion_chatter topic and the subscriber was the flag topic for the photo_click node specifying the turtlebot number to be selected for clicking the photo. The message type for the flag topic was std_msgs/Int8.
- (iii) Photo_click.rapp file to give the description of the Rapp and specify the launch file, interface file and other parameters for the Rapp.

I also created similar Rapps for the emotion_talker and emotion_listener nodes for checking the communication between the master and the clients and debugging.

After creating all the Rapps, I included an export tag in the package.xml file of the Rapp package specifying the location of each .rapp file in the package.

II. Launching the complete framework

For this progress review, we intended to show multiple robots clicking the photos of the person of interest by selecting the robot with the highest smiling percentage.

So, we decided to follow the following steps to achieve this goal:

- (i) Setting one laptop as the system master where the following the command is launched:
Roslaunch turtlebot_concert concert.launch
- (ii) Launching concert clients on the three Chromebooks connected to the turtlebots:

Roslaunch turtlebot_bringup concert_client.launch
 robot_name:=magical_undersea_creature robot_unique_name:=false
 Here, each robot gets a new name (magical_undersea_creature in this case) in the global namespace for a given network.

(iii) Doing ros service call on the master using the following command:
 rosservice call /magical_undersea_creature/start_rapp
 my_rapps/photo_click [{"remap_from: /flag, remap_to: /magical_undersea_creature/flag},{ remap_from: /emotion_chatter, remap_to: /magical_undersea_creature/emotion_chatter}"] []

(iv) Launching the Intraface node on each of the client

(v) Launching the best expression selector node on the master

The rosgraph for the client is shown in figure 1. The figure 2 shows the two camera feeds getting the image of the smiling person but only the one with highest smile percentage clicks the photo.

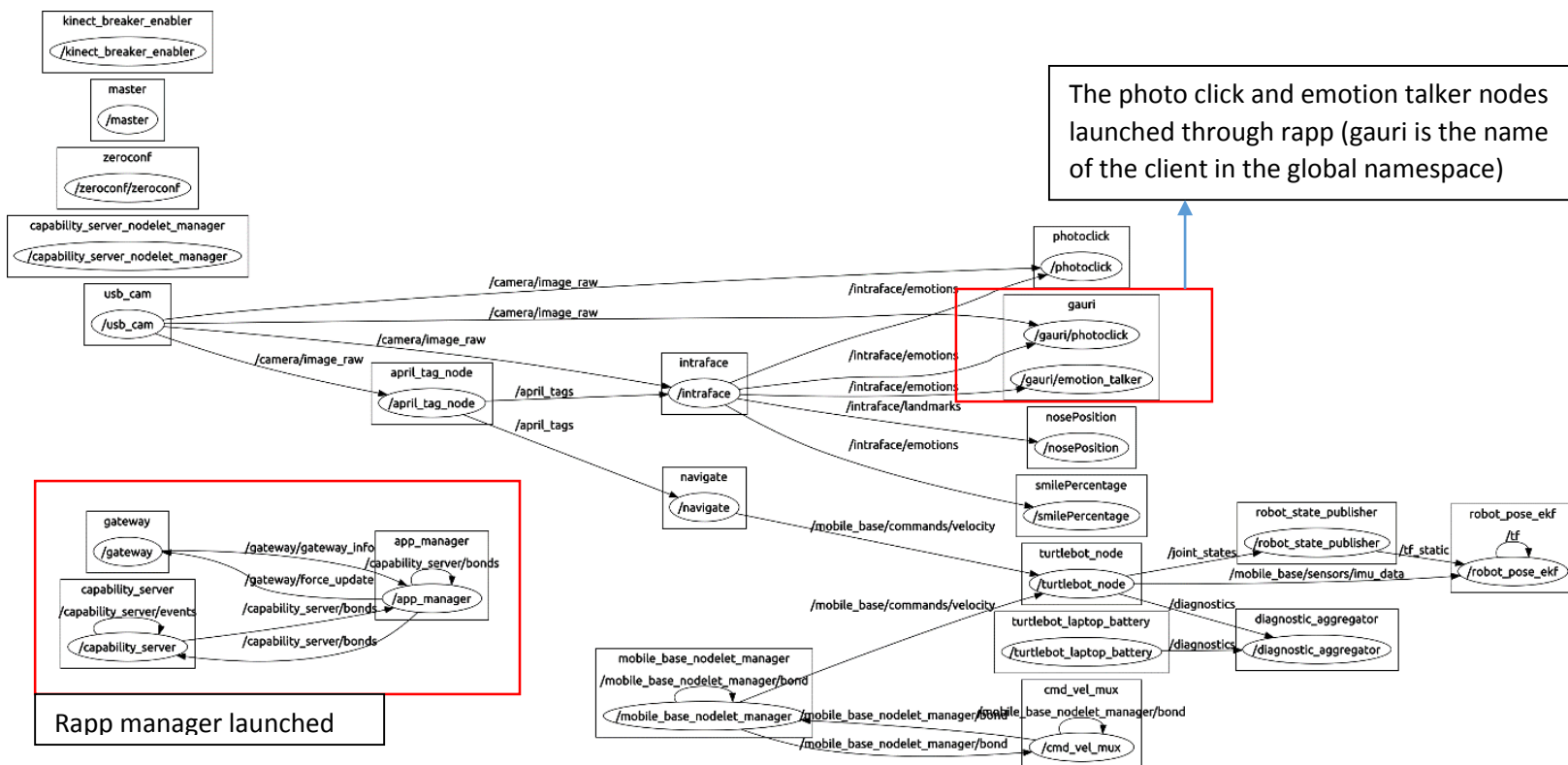


Figure 1 Rosgraph on the client

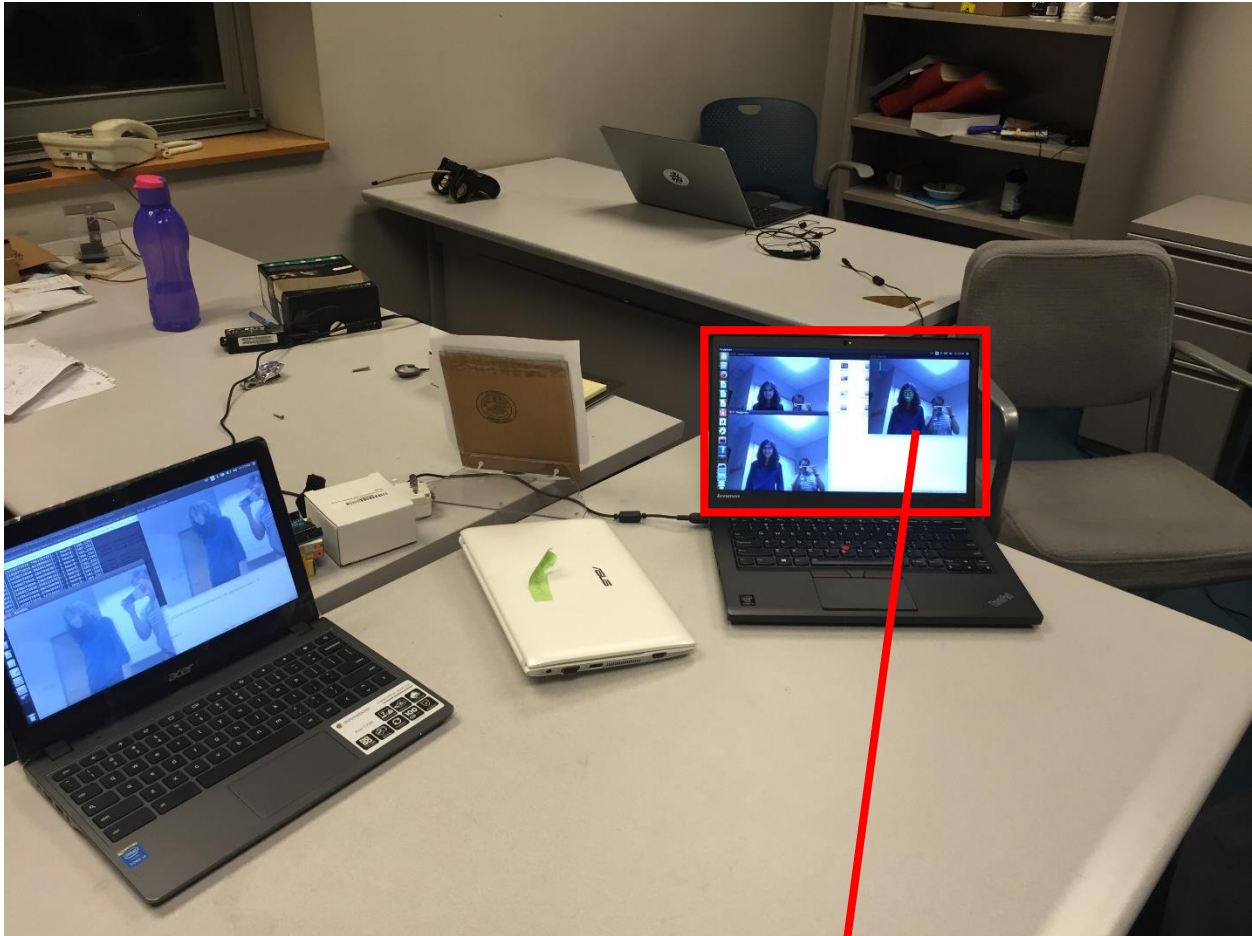
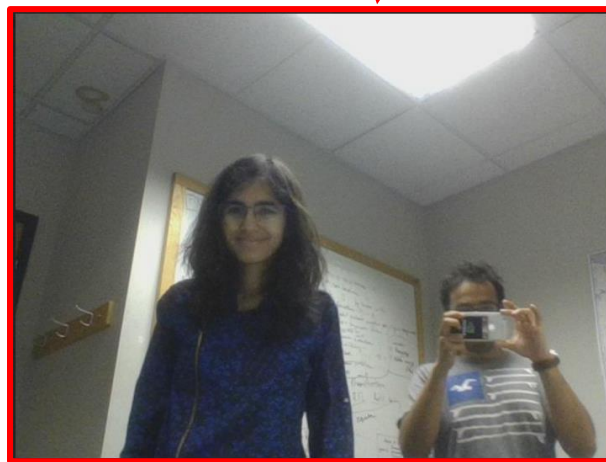


Figure 2 Client with best photo clicks the photo shown in red box



b. Challenges

- I. The first challenge I faced this time occurred at the time of creating my personalized my_rapps package in the clients. The Rapps created by me were working with the rocon_rapps package in the rocon_tutorials workspace but not with the package created by me. It took a lot of time for me to figure out the problem. So, I investigated all the rapp_manager launch files and the concert

launch files in the turtlebot_bringup package. I found out that we need to specify a whitelist of the allowed Rapp packages at the time of launching the files. So, I included the following command in the bash file of each client:

```
export TURTLEBOT_RAPP_PACKAGE_WHITELIST="[rocon_apps,  
turtlebot_rapps, my_rapps]"
```

- II. The major challenge that Jimit and I faced was specifying the remappings between the local and global namespaces. This was quite a tricky task to remap the topic names at different places. After a lot of research and trials, we found out that following remappings are required:

In the client's namespace:

Assuming the topic to be published by the client is my_topic, then in the launch file, remap from my_topic to topic's name in client's namespace, for eg., /my_topic.

Specify the changed topic name in the client's namespace in the interface file.

In the ros service call, remap from client's local workspace to the global namespace, i.e., remap to /magical_undersea_creature/(topic's name in local namespace).

The master can publish to or subscribe directly from the global namespace.

- III. Another challenge I faced was while launching Intraface through the Rapps. Every time I try to launch anything involving a video feed through a Rapp, I am unable to see the video window in the client laptop. I am yet to figure out this issue.

c. Team Work

For this progress review, I collaborated with Jimit to understand the multi-master system and set it up for Intraface. We also took help from Sasanka Nagavalli who explained us the concept of Rapps in ROCON and cleared many of our doubts. Sida worked on to improve the system efficiency by capturing the expressions of only one desired person from a frame having multiple people. Rohit fabricated the new pan-tilt designs and finished the hardware part of the system by setting up the new units on the turtlebots. He also ordered all the required equipment and parts for the system. Tiffany worked on improvising the software implementation of the new pan-tilt unit to achieve software and hardware compatibility.

d. Future Plans

For the next Progress Review, I will be working on the following tasks:

- I. Implement the flocking of the three turtlebots and localizing them using the April Tags and the global cameras set up in the Advanced Agents Lab. I will be collaborating with Jimit and Tiffany for this task.
- II. Create all the required Rapps for the detection sub-system, navigation sub-system and the integrated system.

As a team, we will finish the following goals:

- I. Integrating the detection sub-system for multiple masters with the mechanical sub-system.
- II. Calibrate all the new cameras for the detection sub-system.
- III. Set up the new Chromebooks.
- IV. Include the voice command feature in the system.

e. References

1. Turtlebot ROS tutorials

wiki.ros.org

2. ROCON tutorials for ROS Indigo

wiki.ros.org/rocon/indigo/Guide

3. Gateway model tutorials

[http://redmine.robotconcert.org/projects/multimaster/wiki/Gateway Model](http://redmine.robotconcert.org/projects/multimaster/wiki/Gateway_Model)

4. Sasanka's bitbucket repository

<https://bitbucket.org/snagavallis/turtlebot-swarm-scripts/>