PROGRESS REVIEW 7: ILR06

Richa Varma

Team I

January 28, 2016

Team members: Dorothy Kirlew, Mohak Bhardwaj, Pranav Maheshwari, Shivam Gautam

1. Individual Progress

At the beginning of the Spring semester, the team distributed remaining work into work packages with technical leads for each major subsystem and a schedule for the entire semester was put in place. Having worked on the communication subsystem last semester, it was reasonable for me to continue work on the same as this subsystem is the basis for our collaborative network of vehicles and should be up and functional as soon as possible.

The two tasks I worked on are:
1. Mesh network of multiple Xbee devices
2. Initial design of parking lot

1. Mesh network of multiple Xbee devices

Dorothy and I worked on the communication/collaboration subsystem. The starting point was defining a logic sequence to handle messages being sent over the network. Essentially, the input to the collaboration node should be user commands and the output should be the state of the parking lot to be further processed by the multi-agent planner and used as when required.
It was important to first define a generic message format and then identify message types for handling different states.

Vehicle ID
Message Type
Data Length
Data
CRC

Table 1. Message Format

Message sent over Xbees are:

a. **HELLO**: When a new vehicle enters the parking lot, it instantaneously becomes a part of the mesh network. However, to ensure a methodical approach for self-identification and registration, it sends out a HELLO message that is received by all of the other vehicles in the lot. This provides a way for the already present vehicles to know the formal identity of the new member.
b. **INTRO**: On receiving a HELLO message, the vehicle that was last in queue (and had already identified itself in the network before the newcomer) sends out an INTRO message. None of the other vehicles send the INTRO except the last in queue to

prevent flooding the network. They simply update their database with the identity of the new member.

c. **UPDATE**: Once the incoming vehicle choses a spot or an outgoing vehicle exits its spot, an UPDATE message is sent over the network notifying other vehicles that a spot is occupied/empty. Every vehicle updates its dictionary with the appropriate spot ID against the vehicle ID.
d. **GOODBYE**: On receiving a return command from the user, an outgoing vehicle sends out a GOODBYE message, thereby releasing its spot. All vehicles now remove the exiting vehicle from their dictionaries.

ROS messages:

a. **PARK**: On joining the entry queue (after receiving 'Park' command from the android app), and after the initial identification is complete, the new vehicle obtains a copy of the existing state of the parking lot in the form of a dictionary of vehicle IDs and corresponding spot IDs. It passes on this state to the multi-agent planner node as a ROS message 'PARK'. The planner interprets the spots to be empty or occupied and calculates the best one by associating a cost with each spot considering its distance from the exit and other factors.
b. **RETURN**: On receiving a 'Return' command from the app, the collaboration node directly sends the coordinates of the EXIT to the navigation node as a ROS message 'RETURN'. Using the exit as its final waypoint, the platform navigates to the exit and leaves the parking lot.
c. **UPDATE**: This message is sent by the multi-agent planner to the collaboration node after it decides the optimal spot which the vehicle should park in. The collaboration node updates the latest dictionary it had received in the INTRO message and appends its own vehicle ID and corresponding spot ID. It then sends the UPDATE message over Xbee (described above) to update the state of the parking lot.

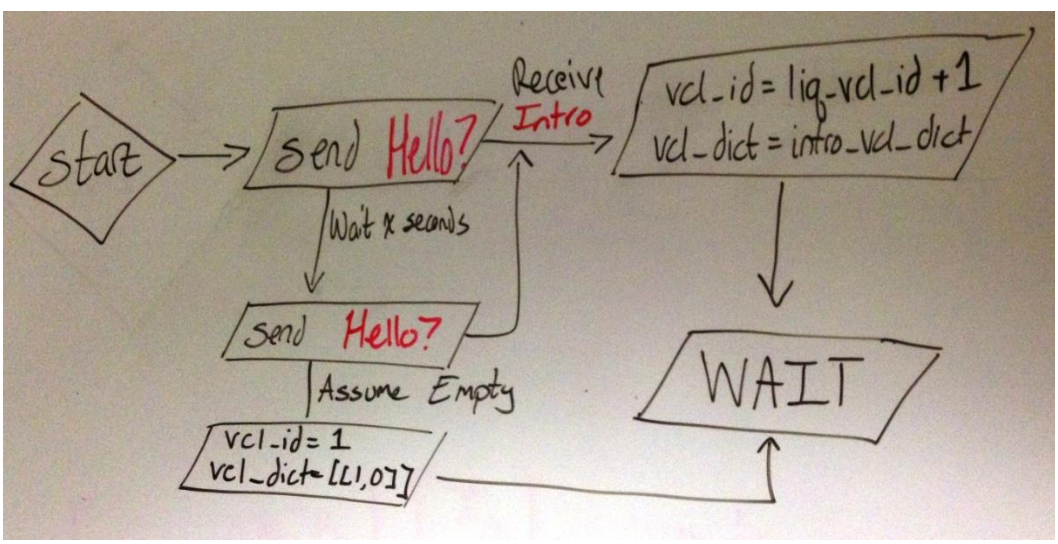

Figure 1: Communication Initiation Sequence (Figure Credits: Dorothy)

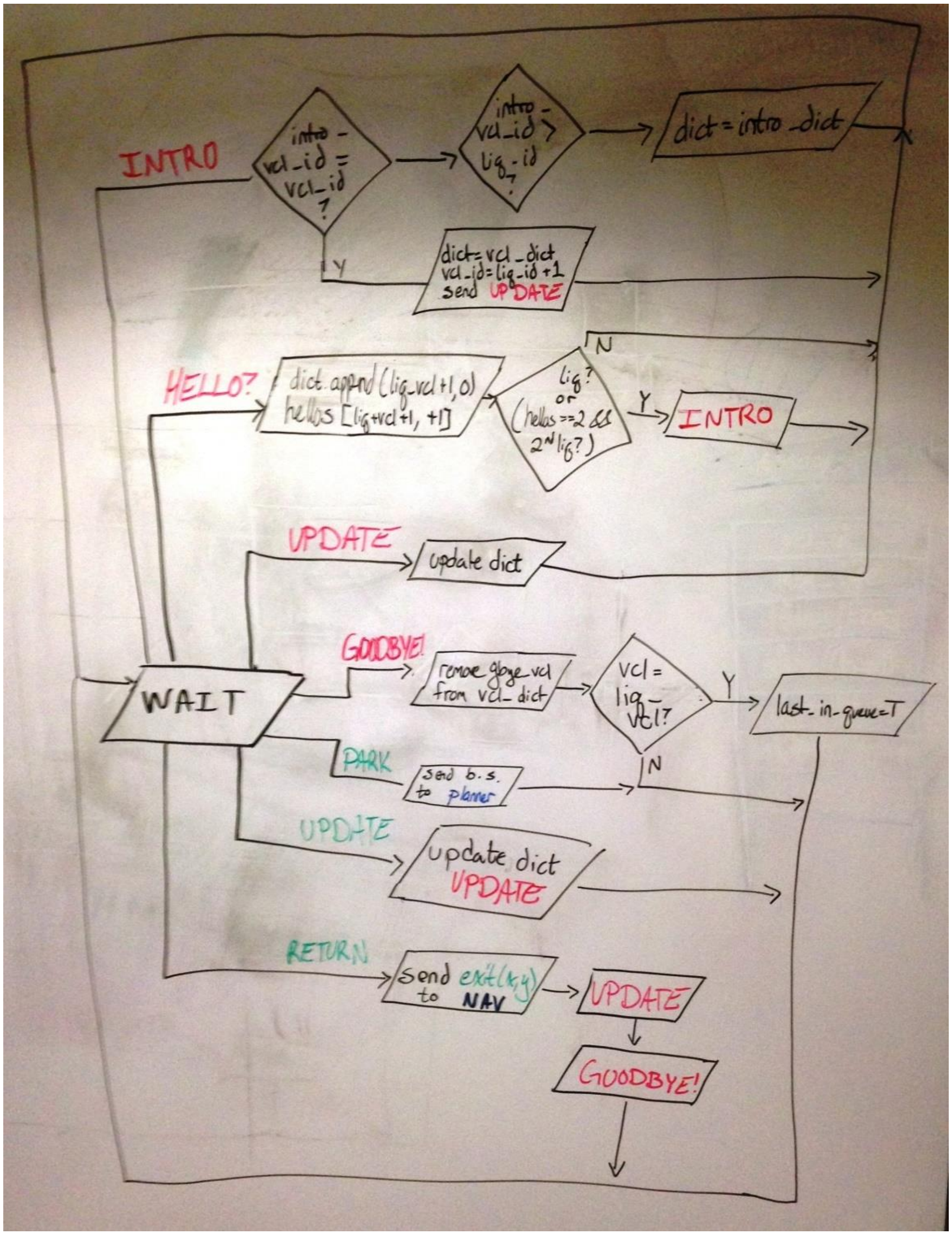

Figure 2: Message Handling Sequence, where green denotes ROS messages and red denotes XBee messages
(Figure Credits: Dorothy)

While brainstorming, Dorothy and I discussed various scenarios and tried to make our message handling as robust as possible. We presented a rough routine skeleton to the team during our meeting, and Shivam pointed out that we have a single point of failure in case the last in queue vehicle failed (did not receive HELLO or dropped out of the network). Hence, we came up with an error handling sequence for such a scenario, where the second-to-last vehicle will send the INTRO if it receives HELLO more than twice in a row from the same vehicle. This is depicted in the figure.

The next step was implementing a small portion of the routine to test mesh network functionality and to lay a foundation for incrementally building up on the code.
Last semester, I had used the python construct library for building and parsing data packets. The best way to start seemed like the basic initiation sequence, i.e. the exchange of HELLO and INTRO messages. I wrote a python script that achieved the following:

A newcomer sends out a HELLO message that is received by the other two vehicles already parked in the lot. On receiving HELLO, the last in queue sends out the INTRO message. The INTRO is ignored by others and is only interpreted by the newcomer. In this case, vehicles 1 and 2 are already parked in the lot. A new vehicle with ID=3 enters and sends out HELLO. Being last in queue, vehicle 2 sends out the INTRO which is received by 3. Now, we make vehicle 2 exit the lot, sending a GOODBYE message. Vehicle 1 interprets this message coming from the l.i.q and being second-to-last, it changes its last in queue flag to 'True' and starts sending INTRO messages to vehicle 3.Through this sequence, we tested the mesh network and have a now have a base for building up our algorithm.

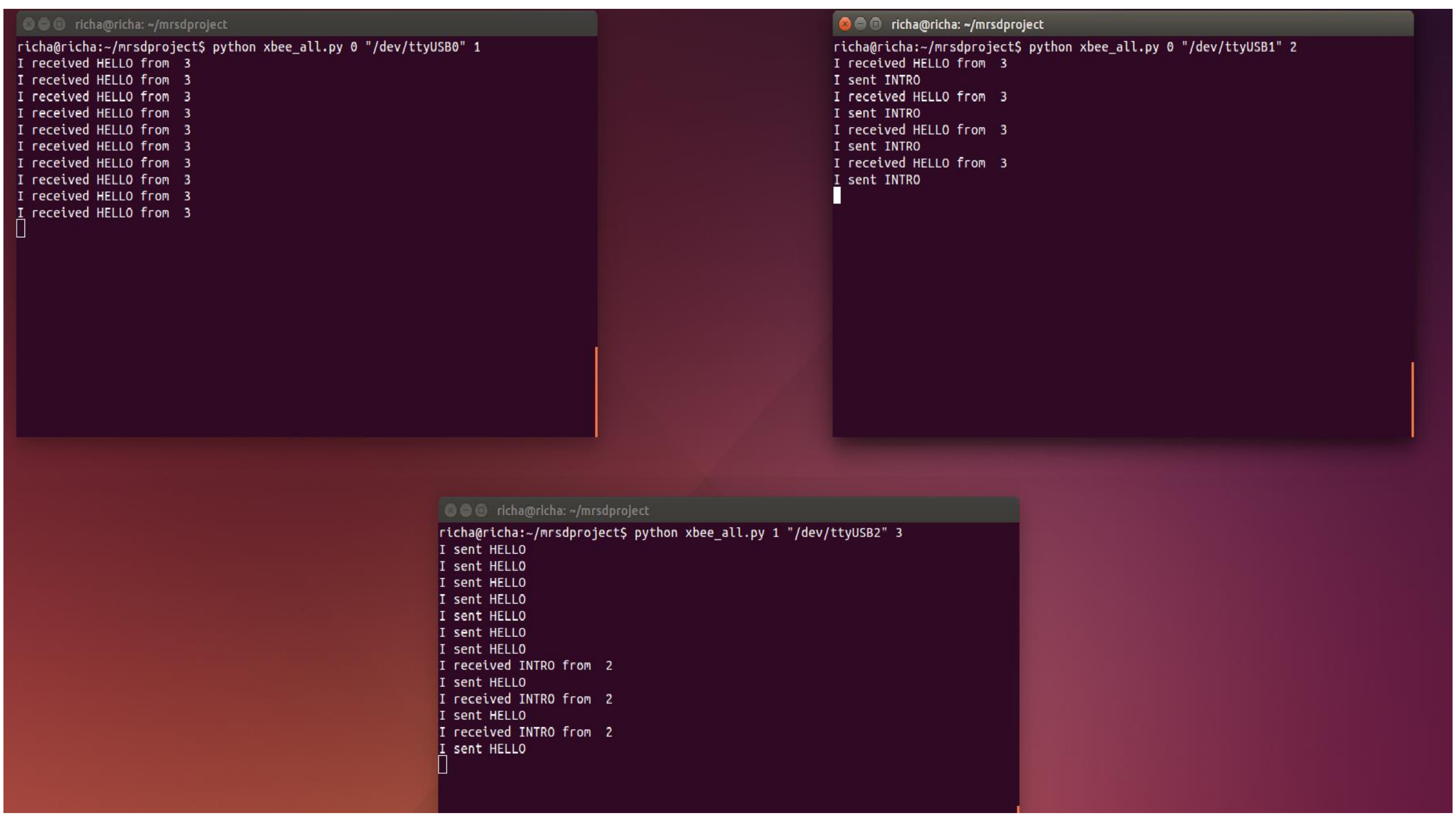

Figure 3. Exchange of messages when all 3 vehicles are in parking lot

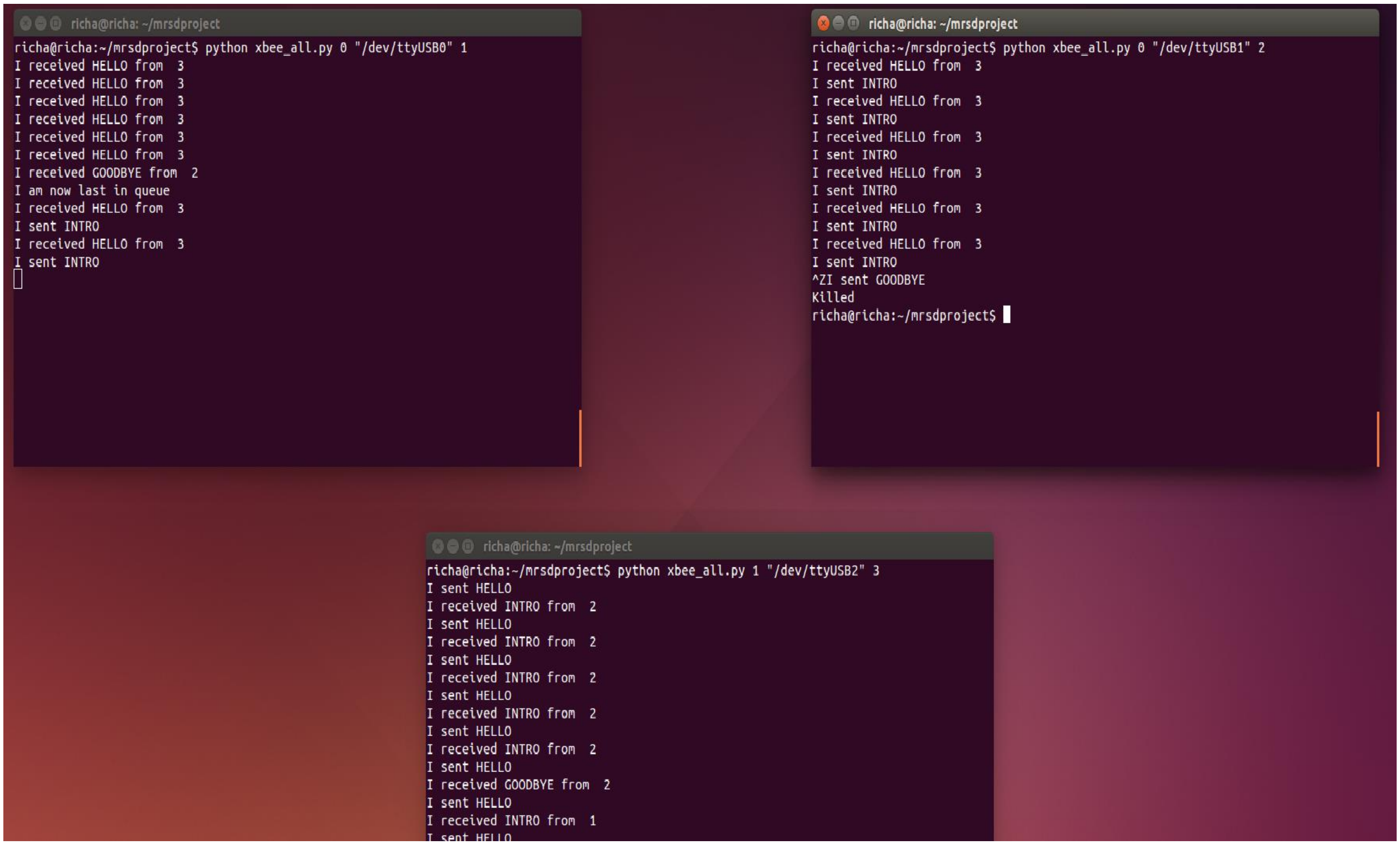

Figure 4. Exchange of messages when vehicle 2 leaves

2. Design of parking lot

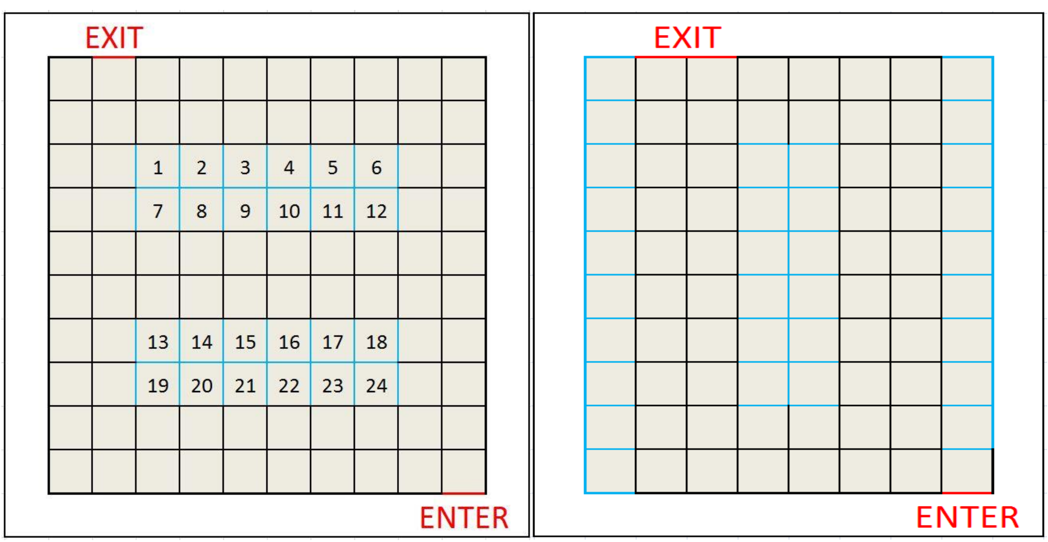

Figure 5. Parking lot layout options (Figure Credits: Dorothy)

Dorothy and I also worked on the initial layout of the parking lot. In accordance to our performance requirements, we need to have a buffer area around the vehicle while it is in the spot. The Oculus Prime platform has a size of 30cmX30cm. Hence, we decided on a cell area of 0.5m2 which led to 24 parking spots and three driving lanes on the lot. We discussed possible designs out of which the two shown in the figure were the final choices. We presented them to the team and decided to go ahead with the one above with spot numbers. Since mapping required rich features in the environment, the next step would be putting walls and corners around the lot and finalizing the design based on input from Mohak, who will be working on map generation.

1. Challenges

Being on schedule has always been a major challenge but this semester, with three heavy technical courses, it is becoming increasingly difficult to be on track. As far as the work up to this progress review is concerned, I did not face any major technical challenges. Since I built up on the code I had written during the Fall, things went smoothly. But I do anticipate some issues to come up as the code gets more complex and we find scenarios that our routine does not handle.

2. Teamwork

For this Progress Review, I worked with Dorothy on the collaboration node. We made good progress and two perspectives made it easier to identify loopholes and errors, which will prove beneficial in the long run. Dorothy additionally tested the android app and ensured it is good enough for the final demo. Pranav worked on the visualization tool and began work on the navigation subsystem, got familiarized with ROS Navigation stack. Due to the Xtion arriving late, Mohak was unable to work on the Mapping subsystem as much as he planned but was instead able to work on the Multi-Agent Planning subsystem. Shivam was unable to work on Obstacle Detection for the same reason, so he instead worked on redesigning the power distribution board PCB.

3. Plans

The team is on schedule with all the tasks and has planned to start integration after February 7th, after the additional mobile platforms arrive. Following are the tasks planned out for future weeks:

1. Android App (Dorothy)
    a. This is already completed
2. P2P Communication (Richa and Dorothy)
    a. Completely code the routine
    b. Test code with 2 and 3 XBees in multiple scenarios
3. Locomotion (Pranav)
    a. Define routine
    b. Test localization
4. Mobile Platforms
    a. CAD and print IR mounts (Dorothy and Pranav)
    b. Electrical assembly of platforms (Shivam and Richa)

5. Obstacle Detection and Perception (Shivam)
    a. Extensive testing with laser rangefinders and integration in ROS
6. Mapping of Parking Lot (Mohak)
    a. Map test area
    b. Map mock parking lot
7. Creating Parking Lot (Dorothy)
    a. Plan parking lot with respect to Xtion requirements and restrictions
    b. Integrate parking lot plan and Xtion restrictions to fully design parking lot
    c. Fabricate and assemble parking lot
8. Multi-Agent Planning (Mohak)
    a. Code and test
9. User Interface (Pranav)
    a. Completely code and test in multiple scenarios
    b. Integrate with XBees to receive appropriate updates