

Colony Intersection Handling Specification: Linked List

February 21, 2011

1 Overview

The linked list intersection handling algorithm uses a doubly linked list to implement a queue. The bots are connected in this list via wireless pings and acknowledgements.

2 Specifications

2.1 enterIntersectionQueue

When a bot enters an intersection, it pings and waits for a reply. If there is no reply, then the bot is the first robot in the list. Otherwise, the last bot in the list will reply and the entering bot is now the last bot.

Example: Suppose bot 10 enters the intersection. It pings and waits for a response. Bot 7 is in the queue, and it pings back and points its back pointer to bot 10. Bot 10 points its front pointer to bot 7 and is the last bot in the queue.

2.2 waitInIntersectionQueue

When the bot is in the middle of the queue, it waits for the bot in front of it (identified using the front pointer) to leave the queue (See next section). When the bot is the end of the queue, then it will also listen for incoming bots (See previous section). While waiting, the bot pings the bot in front of it to make sure the previous bot is still present (for robustness purposes).

2.3 driveThroughIntersection

When a bot leaves the intersection, it sends a packet out announcing its departure. The next bot in line (via use of its front pointer) will acknowledge the leaving bot and sets its front pointer to null, making it the front bot.

To implement: Before the bot leaves the intersection, it will check if it can

move through the intersection without colliding with other bots. Ideally, this means that two bots can drive through the intersection at once provided that they are going in opposite directions.