Our main algorithm uses the destination of the vehicle the packet is trying to reach as a method for routing the packet. Whenever a vehicle reaches its destination it recalculates a new destination and shares that information with the rest of the world. This information is shared by transmitting an update packet with the vehicles new destination and from what position it is starting from. This packet transmission is sent to all vehicles in the system via a flood algorithm. Each vehicle contains an array of vehicle location information structs which contain the same information sent in the update packet. Upon receiving an update packet that vehicle will updates its list of vehicle destination and starting locations.

When Vehicle A is given a packet whose destination is Vehicle B the Destination Search algorithm is called. Vehicle A will first check its list of surrounding vehicles(Which is updated every tick) to see if Vehicle B is within range, if that is the case, Vehicle A will throw the packet to Vehicle B and get rid of that packet. If Vehicle B is not in Range then Vehicle A will look at its array of vehicle locations to see if it has any information on Vehicle B. If Vehicle B is in its list then Vehicle A will take Vehicle B's destination and compare it to Vehicle A's current location to determine which direction would be ideal to attempt to throw the packet for it to intersect with Vehicle B. Once Vehicle A has an ideal direction it cycles through all vehicles in its immediate vicinity and assigns each one a score of how ideal it would be to throw to that vehicle. The score is based off its current direction, its current destination and its position in relation to the ideal direction to throw calculated earlier. Vehicle A also gives itself a score to determine if the best choice is to not throw the packet at all. After all vehicles are considered the packet is thrown to the vehicle that received the highest score. And the process repeats with the new vehicle.