Manipulation of Nodes in a Connected Car Network

Henry Hexmoor[1], Guy Fraker[2]
[1]Computer Science Department
Southern Illinois University
Carbondale, IL 62901, USA
[2]AutonomousStuff.com

Abstract - *The future of electronic, connected, and networked automobiles is upon us. This new technology reshaping the way we drive and communicate on motorways. The technology cars maintain connections and communicate via Wi-Fi. The Connected cars initiative envisions cars that form a dynamic social network. Each car exchanges local information such as speed and travel direction with peers. Other than increased safety and traffic improvement, cars are anonymous in the network. In this network, it is possible to use neighborhood information to blend in and evade traffic from law enforcement authority. For example, if there are a hundred cars driving in unison, it is difficult to apprehend speeding vehicles. If the police vehicle is interested in a specific car, it needs to enter the network and positions itself behind the target car. This project explores techniques for infiltrating a connected car network to form ties with a specific node.*

Keywords: *connected cars, networks, law enforcement, multiagent simulation, netlogo*

1.0 Introduction

A connected cars network is a network of IP equipped automobiles. This enables a vehicle to share data with other devices both inside and outside the vehicle. Connected vehicles can vastly improve the transportation system in the areas of safety, mobility and environment by providing connectivity among the vehicles to enable crash prevention and infrastructure. Wireless devices will provide continual real-time connectivity to all system users. It also provides connectivity among the vehicles and the infrastructure to enable safety mobility and environmental benefits. Connected cars network relies on wireless ad-hoc networks and Wi-Fi direct technologies [2][3][4][6][7]. It is a decentralized type of wireless network where each node participates in routing by forwarding data for other nodes, and so the determination of which nodes forward data dynamically is made based on the network topology. Wi-Fi direct is independent of access point to communicate with a mobile agent [5]. It works by embedding a limited wireless access point into the devices [1]. Each vehicle is linked to a node in a wireless ad-hoc network. Whenever other nodes come in communication range of a node, it uses Wi-Fi direct technology to communicate with that node. It sends its vehicle ID number to the other node. It also receives the same information from other nodes. The goal of this project is to enable law a enforcement vehicle to scan for and position itself near vehicles that violate the local speed limit. The vehicle will then search the database for the ID number, which contains an entity is in the format of <id, location, speed, distance from a police vehicle, ID of the alleged suspect vehicle>. If the vehicle finds a speeding vehicle or intruder vehicle, it stores the offender vehicle ID and forwards this information to its neighbor node and to the neighbor police node. If a police node is not a neighbor of a node then it compares the shortest distance to police by comparing the hop-distance to police through other nodes. When a police node receives the tuple <ID, location, direction of a speeding/intruder node> through a network of nodes or a single node then it starts pursuing the intruder node. We used the concepts of Wireless ad-hoc network and Wi-Fi Direct technology. Ours is a decentralized Wireless network where each node participates in routing information by forwarding data to other nodes. Hence, the determination of which nodes will forward data uses the network structure. Wi-Fi direct does not depend on access points for communication with a mobile agent. It works by embedding a limited wireless access point into the devices. Each vehicle is linked to a node in a wireless ad-hoc network. Whenever other node comes in communication radius of a node, it uses Wi-Fi direct technology to communicate with that node. It sends its vehicle ID number to the other node. It also receives the same information from other node. The vehicle will then search the database for the ID number, which contains an entity is in the format of <car id, location, speed, its distance from a police vehicle and the ID of the alleged suspect vehicle>. If the vehicle finds a speeding vehicle or an intruder vehicle, it stores the offender vehicle ID and forwards this information to its neighbor node and the neighbor police node. If a police node is not a neighbor of a node then it compares the shortest distance to police by comparing the hop-distance to police through other nodes.

2.0 System Analysis

Every node broadcasts car id to its immediate neighbors, which are in its Wi-Fi range. A single node sends and receives a set of information from its neighbors. The information includes the ID number. Every agent has an ID number [1]. The number distinguishes it from other agent. Like a registration number of a vehicle, a node has a unique ID number. In Wi-Fi environment, this serves as a MAC address for a node in the network. Whenever a vehicle enters into the range of the network it automatically connects with the network as shown in Figure 1. In this design, the wireless network is always available.

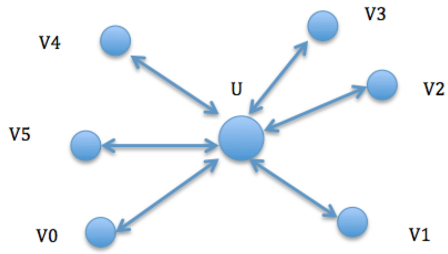

Fig 1. Illustrating how the connected cars system work by taking an example network

A mobile agent changes it place but ID number remains the same. Every single agent sends its ID number to its immediate neighbors and receives its neighbors ID number as well. If a node finds an intruder node or speeding node as its immediate neighbor then it stores the information of the intruder node as:
1. Speed: Current speed of the node.
2. Location: Current location of a node.
3. Target Node: If a node finds an intruder node or speeding node as its immediate neighbor then it stores the information of the intruder node as Target Node. And in later broadcast of the information along with other information it sends the Target Node information to its neighbors. If a neighbor node finds the information of the Target Node from its neighbor then it also updates its own information about Target Node with this and looks for the shortest distance to police to forward Target Node information.
4. Hop distance to police: In this proposed design, all nodes are mobile. Instead of maintaining static information about hop distance to police, the routing table always updates itself. Suppose a mobile node having a police as its immediate neighbor. Then its hop distance to police is 1. After a while the node changes its place where police is not 1. Other than itself an immediate neighbor of its own is connected with a police. Then the hop distance to police will be 2. So a mobile agent in this system always updates its hop distance to police.

2.1 Simulation Setup

• Every vehicle (i.e., agent) node in the network has its own database table in the MySQL database.
• Technically, a Netlogo agent cannot forward its information table to another agent [8].
• In a simulation of Netlogo [8], there are multiple agent nodes with multiple interactions between them which needs database update many times a second. So, connecting to MySQL is somehow technically convenient but database triggering, fetching, storing take extra overhead to maintain. As shown in the Figure 2, SQL database is connected to Netlogo by using extensions keyword [9].

NetLogo-sql (or NetLogo SQL Wrapper) is a JDBC based extension for the NetLogo modeling environment to access databases using SQL queries. It allows NetLogo modeling environment to use a relational database by providing better and easier use of interface for storing and retrieving data than currently provided by NetLogo, using lists and files during the execution of a model simulation and speed up the execution of models. The SQL Wrapper extension provides a way for NetLogo simulations to access existing data from, or update or inserts data into a relational database with an existing database schema. As shown in Figure 2, there are four components to enable this:
- NetLogo with its extension-mechanism;
- The SQL Wrapper implementing the extension-mechanism;
- A database access layer (i.e., JDBC and a suitable driver);
- A database system.

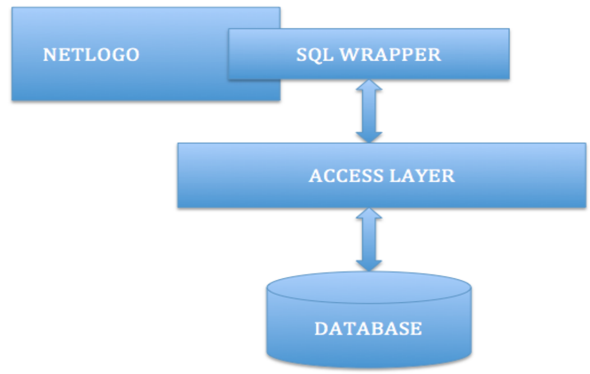

Fig 2. Sql wrapper acts as a bridge between NetLogo and the database

The NetLogo layer is concerned with accepting commands from NetLogo and returning data. Similarly, the database layer is concerned with communicating with the database. To separate these, a third layer is provided as a bridge between these two: The wrapper layer. This layer provides mapping between Netlogo data primitives and SQL primitives, and other mappings that are translations from Netlogo to SQL and vice versa. The context layer takes care of resource management and additional configuration. In this case, the connection pooling as implemented using an external software library called BoneCP. To allow the Netlogo user to use the connection pool, specific command syntax will added to enable and configure it when needed. All layers are depicted in Figure 3.

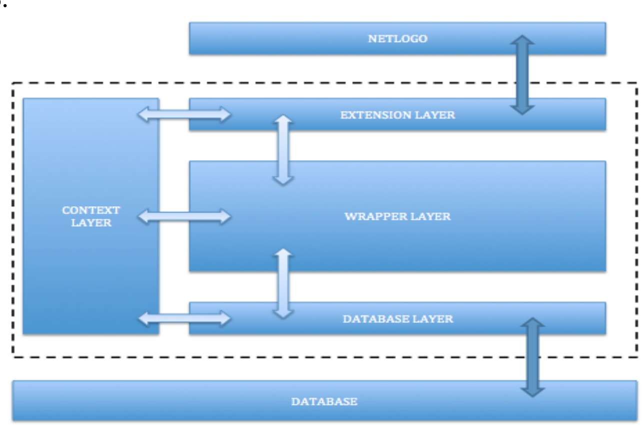

Fig 3. The internal architecture of sql wrapper has four layers

3.0 Implementation and Results

For validating the connected vehicles concept, we implemented a prototype using Netlogo 5.0.1 simulation rapid prototyping platform.

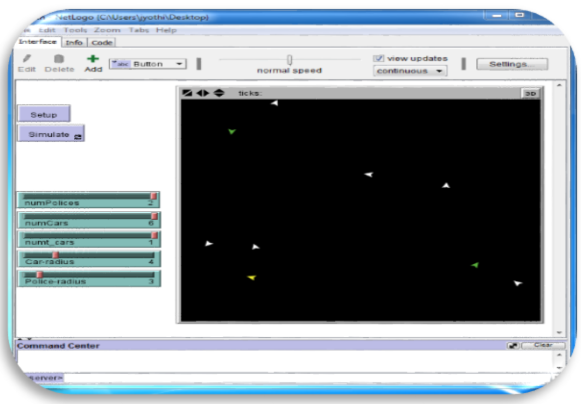

Fig 4. The interface of the present project

Netlogo library tools are included for the interface as shown in Figures 4, 5, and 6. In order to generate result for various input environment, a controller from number of simple vehicle, intruder vehicle and police vehicle are incorporated. A slider tool is provided to adjust the radius of the commutation range for police cars. We monitored the simulation output in multiple case studies. The range of adjustment tools includes the following:

1. numPolice vehicless: Number of police for the simulation environment
2. numCars: Number of cars for the simulation environment
3. numt_cars: Number of target in the simulation environment, and
4. Police-radius: Police communication range with its neighbors.

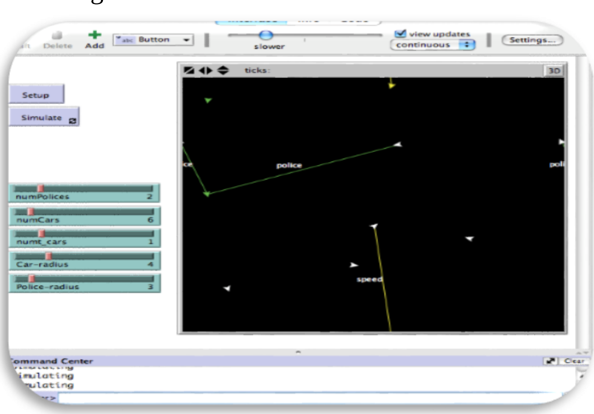

Fig 5. Agents connections in the data base and exchange data

In Figure 5 at a simulation moment, we observe wireless connection among nodes. Police, regular vehicles and intruder car are connected with one another and form a mobile network of information with neighbors. When an agent connecting with high speed agent then we name the link as ‘speed’. When connecting with police then we name the link ‘police’. A node connects with a police node and sends its information to the police node. Upon receipt, a police node searches for a target node information in the set. If a police node finds information about a target node, then it changes its direction toward the target Node’s direction and location and starts to chase the target node. But in the simulation, target node’s speed is much larger than the police’s. When a police node approaches the target node within its sensing radius distance, then it connects with the target node with a link. It apprehends the target node agent.

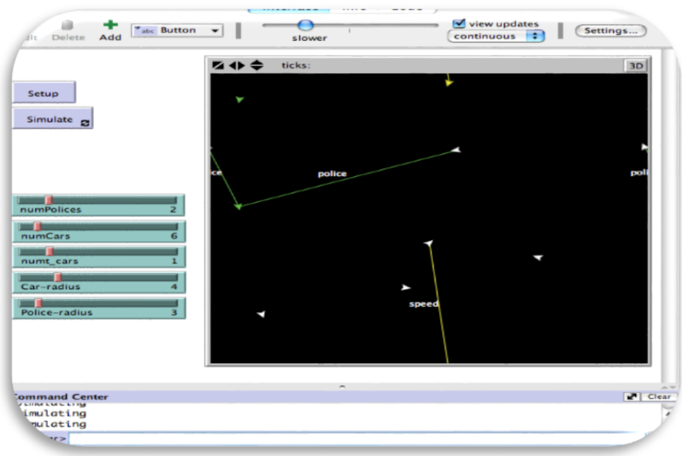

Fig 6. Agents connecting in the database and exchanging data

In Figure 6 at a simulation moment, we can see the wireless connection between agent nodes. Police, vehicles and intruder car are connecting with one another and forming a mobile network of information with its neighbor. When a white agent node connecting with high-speed yellow agent then naming the link as ‘speed’. When connecting with green police then naming the link ‘police’. A white node connects with a green police node and sends its information to the police node. On receiving, a police node looks searches for Target Node information in the set. If a police node finds information about a Target node, then it immediately changes its direction towards Target Node direction and location and starts chasing the Target Node. However, in the simulation Target Node speed is much higher than the police vehicle. When a green police node closes to the Target Node radius distance then it connects with Target Node with a Red link. The police vehicle apprehends the Target Node, yellow agent. The database is continually updated with the status of the vehicles. The target node is set to the highest speed car and the hop distance to the police is updated along with them. When the vehicle lost its connection with other vehicle, immediately the database is updated and the values of that particular vehicle are updated.

4.0 Conclusions

Connected car technology is a rather novel arena that enables exploration of possibilistes for innovation. These are useful for networked cars on the road and the space. Soon, there will be large networks of cars that can be seen moving as liquids yielding car lakes. As such, we need to use network properties to analyze dynamics of car networks. These can be useful for individual nodes as in measuring latency of effects of events reaching a node of interest. We have implemented a prototype and initiated navigation within it. This work creates a foundation for future studies. Managing networkwd cars as lakes to be monitored and directed is a promising direction for exploration.

Acknowledgement: Ms. Jyothi Vaddula implemented the netlogo model

References

[1] Woolridge, M. & Wooldridge, J. M., (2001) , Introduction to Multi Agent Systems, New York,NY, John Wiley & sons,
[2] Aruba networks (2011) , Next Generation Wireless Mesh Network
[3] Camps-Mur, D., Perez-Cost, X., & Sallent-Ribes, S., (2011), Designing Energy Efficient Access Points with Wi-Fi Direct
[4] Frodigh, M., Johansson, P., & Larsson, P., (2000), Wireless ad hoc networking -The art of networking without a network
[5] Aneiba, A., & Rees, S.J., Mobile Agents Technology And Mobility
[6] Kumar, P.R., Borkar, V., Gupta, P., Kawadia, V., Narayanaswamy, S., Rozovsky, R., Sreenivas, R. S., & Xie, L. (2002), Ad Hoc Wireless Networks :Analysis, Protocols, Architecture andTowards Convergence.
[7] Sanai, D., (2001), Detection of Promiscuous nodes using ARP packets.
[8] Tisue, S., & Wilensky, U., (2004), NetLogo: A Simple Environment for Modeling Complexity.
[9] Blom, J., Quakkelaar, R., & Rotteveel, M., NetLogo SQL Wrapper User manual