COMPUTER SCIENCES AND SOFTWARE ENGINEERING
AUBURN UNVERSITY

COMP 5360/6360
Wireless and Mobile Networks
Spring 2014

Project 1

Semi-Truck Cooperative Adaptive Cruise Control with Efficient Flooding Algorithm in a Vehicular Ad Hoc Network (VANET)

Due: 11:55pm March 20, 2014

1 Objective

The goal of this project is to design and implement an *efficient flooding algorithm over a vehicular ad hoc network (VANET) that will be useful for a semi-truck cooperative adaptive cruise control application.* In your emulation, each vehicle is equipped with a mini PC with a simulated WiFi environment, i.e. the communication range and propagation characteristics (e.g. packet loss) will model actual WiFi links set in ad-hoc (IBSS) mode. In this *emulation environment*, you may emulate the underlying wireless link layer with exchange of UDP datagrams between two neighboring hosts through AF_INET sockets. However, the protocol you will implement can be ported to an actual network layer. In this project, sensor and control packets will be transmitted over an emulation of the wireless vehicular ad-hoc networks (VANET) that use an efficient flooding ad-hoc routing protocols. The emulation of the mobile VANET is implemented over a fixed network.

2 Background

Wireless communication between semi-trucks can enable these semi-trucks to platoon with each other in the highways, resulting in several benefits, including fuel saving, efficient use of limited highway capacities and highway safety. The aerodynamics of the truck platooning configurations will reduce highway vehicle drags of bodies resulting in lower fuel consumption. Critical sensor and control information must be exchanged between the semi-trucks in order to enable them to automatically maintain the platooning of the several trucks that are traveling together, where the lead truck will be followed closely by a train of other following trucks. The sensor data exchanged between trucks must contain accurate vehicle information, such as GPS position, velocity, acceleration, yaw, and so forth. These sensor data must also be exchanged at a high rate to enable real-time response between the lead and following trucks. Furthermore, the sensor data must be transmitted with high reliability.

Due to the limited range of the communication device and due to other propagation characteristics, the wireless devices in different platoons of trucks could be spread over a large area forming mobile vehicular ad hoc networks (VANETs). The VANET will be used to forward packets from the lead truck to the truck at the end of the platoon. Vehicular sensor data packets from the lead truck will be forwarded using the VANET routing protocols through multiple hops of the intermediate wireless devices and eventually to the last truck at the other end of the platoon. Sensor data packets from the last vehicle may similarly be forwarded through similar multi-paths in the reverse direction to the lead truck.

3 Requirements

The objective of this project is to implement a flooding routing algorithm over a VANET for supporting multi-hop forwarding of vehicle sensor and control packets for a cooperative adaptive cruise control system that will enable trucks to platoon with each other. Using these programs, a truck will be able to form a platoon with another truck or join into an existing platoon of trucks in the highway. Vehicle sensor data from each truck will be broadcasted to all the other trucks in the platoon in order for them to maintain their platoon configuration.

3.1 Truck Platooning Application

Write an application program that will simulate the control for a truck that will enable it to participate in automatic truck platooning by providing the following functions.

(1) *Form a platoon with another truck.* Two trucks initiate forming a platoon where one will be elected to be the lead truck using some criteria, e.g. high average speed. Once the platoon is formed, the truck platooning control application will ensure that the two trucks will be travelling at the correct speed to maintain the platoon configuration, i.e. as long as the distance between the trucks is within the range of 10 to 20 meters. Assume that the average speed of each truck is 30 meters/sec., where the standard deviation of the speed is about 10 meters/sec. For more realistic dynamic models, you can also assume that the acceleration and deceleration varies from 0 to 1 meter/sec^2.The actual speed of each truck in a platoon may be different.

(2) *Join an existing platoon*. A truck that receives vehicle sensor information from other trucks in the platoon may decide to join the platoon. Once it joins the platoon, it will maintain the platoon configuration as above.

(3) *Leave a truck platoon*. A truck may leave a platoon and need not be bound by the rules for maintaining the platoon configuration.

Assume that all trucks are travelling in one direction in a highway, e.g. East, and that the highway consists of only two lanes in the same direction. Assume that the width of each lane is 5 meters.

In the simulation application above, assume each semi-truck is 25 meters long. You must simulate at least *five* trucks travelling in a platoon. No two semi-trucks may overlap each other in space on the same lane in the highway. For example, consider a scenario where a semi-truck *T* is merging into the middle of an existing platoon that is traveling along the right lane. Then *T* must be traveling on the left lane and there must be sufficient empty space between two trucks in the platoon, say 45 meters, in order for *T* to move to the left

lane to merge into the platoon. Once merged, T must maintain the allowable distance from the front and back truck, i.e. between 10 to 20 meters from either trucks.
In order for a truck to learn about the position, speed and acceleration of other trucks, each truck must transmit its sensor data containing its GPS position, speed and acceleration at a constant period of 10 milliseconds. These sensor data are then forwarded multi-hop using your flooding algorithm.

3.2 Flooding Algorithm for VANETs

You will design and implement the flooding protocol on top of the simple UDP socket interface that emulates the wireless link layer represented by the shaded portion of Figure 1. The shaded portion is not part of this project, but represents protocols that could conceptually be added or substituted at the layer where it is shown.

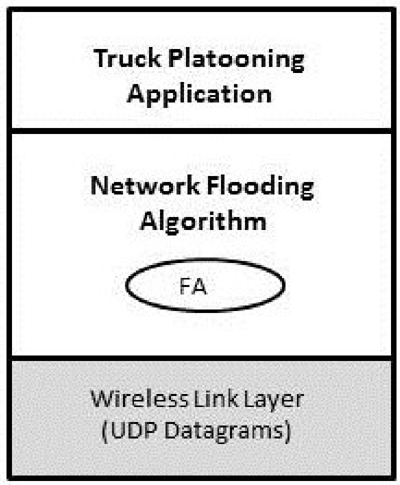

Figure 1. Overview of the VANET Protocol Components

Initially, a source truck node will generate a vehicle sensor packet depending on its dynamics and forwards it to all its neighbor nodes through its FA layer. Eventually the packet will be forwarded through multiple hops and reach all other trucks destination address. The FA layer of the intermediate nodes performs the task of forwarding packets efficiently from one node to another. The packet header must contain at least the following fields:

1. Sequence number
2. Source address
3. Previous hop

One issue in FA forwarding is the addresses in the packet header. Every wireless node in the ad-hoc network has a globally unique 16-bit address (not IP address), which you can define. Since we will not be focusing on unicast or multicast forwarding, the vehicle sensor packet's destination address will contain a special broadcast address. Although typically the higher-half of the address space is reserved for distant network broadcast addresses, you will use only a single network in this project, in which case, all the address bits of the broadcast destination address is set to 1.

The FA packets are limited to 4,096 bytes, including a header. For longer messages, the higher layer protocols or applications must perform the corresponding segmentation and reassembly of the message to/from different packets.

You must calculate the packet drop rate for each link which is proportionate to the square of the distance. Develop a function to compute the packet drop rate. When the error rate is above a certain threshold value, consider the node is out of range. Set your packet loss function so that the range of each node's transmission is about 100 meters. The error rate should capture the propagation loss (optional to include the effect of slow fading and fast fading). You can calculate the distance between nodes based on the current location.

Each router will run the flooding protocol as follows. When a node with address j receives a packet, it first checks the source address i of that packet. If $i = j$, then the node will discard the packet, since it is the source of the packet. Otherwise it checks in its *cache table* the largest sequence number of broadcast packets that it has received from source i. (Each FA layer maintains a *cache table of broadcast packets* that it has received from each source with the largest sequence number of packets that it has received from it up to that point, i.e. the cache table contains information on flooding packets that the node has forwarded previously.) If that sequence number in the cache table is smaller than the sequence number in the new packet, then that packet is new and *needs to be forwarded.* The router j then updates the cache table with the largest sequence number that it has received from i. It then forwards a copy of the packet to *each* of its neighbors that is not the neighbor k that it received the packet from. The neighbor k can be determined from the "Previous Hop" field specified in the packet header. Eventually the packet is sent to all reachable wireless nodes.

Some of the design issues for FA that you must address are as follows:

1. FA packet header
2. Forwarding mechanism for flooding packets
3. Structure of cache tables
4. Memory management of packet buffers

3.3 Network Configuration

Each network node will be emulated by a separate execution of your program in one of the tux computers. Nodes will run on a subset of the COE instructional Linux workstations in Davis 123, using the computers tux175 – tux198 with each node corresponding to a (machine IP address, UDP port) pair. Use only port numbers that are assigned to you. Therefore, each execution of your program will represent a virtual node communicating with other virtual nodes, each through its own UDP port. There may be many such virtual nodes running on a single workstation or running on several workstations.

A configuration file will specify the virtual nodes in the network and the links between them. Your program will take the configuration file as a command line argument. Then, the configuration file will have to be parsed and a copy of your program will be executed in each virtual node. The format of the network configuration file is shown next, while the topology of the network represented by that file is shown in Figure 2. The x- and y-coordinates of each node is specified preceding the link keyword. For example, the coordinate of Node 1 is (50, 120).

Node 1 tux175, 10010 50 120 links 2
Node 2 tux175, 10011 80 120 links 1 3 4
Node 3 tux180, 10010 180 170 links 2 4
Node 4 tux180, 10011 150 60 links 2 3

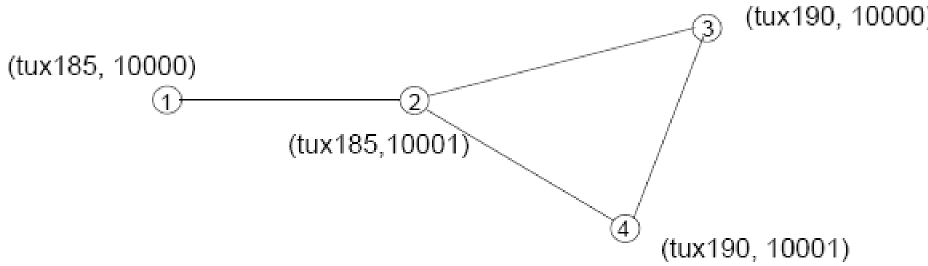

Figure 2. Network Topology

To emulate mobility of hosts, your program running in each virtual node must constantly monitor any change in the configuration file. When a mobile host moves to another location, its link to other hosts in the previous location may be disconnected while a new link may be established to yet other hosts in the new location. The change in the configuration file can be manually done or automatically updated by a program that accounts for the dynamics of the trucks with some randomized speed. The configuration file may also contain location information of each node. Each node periodically read the configuration file to determine the set of neighbors that it can reach through a wireless links and their new locations.

4 Testing

Your truck platooning simulation application programs must run over the virtual VANET which must execute correctly in the College of Engineering Linux tux computers in Davis 123.

The truck platooning application that broadcasts vehicle sensor packets to other trucks will make use of the network flooding algorithm functions that you have implemented above. Make sure that all wireless nodes are executed on different computers. The truck platooning application must allow all trucks to communicate with each other, whereby each truck node must use a different tux computer.

You must run this application with at least 5 wireless nodes (representing the semi-truck wireless systems). These wireless nodes are distributed in any area large enough for the trucks to travel over a 5 minutes simulation period. At the beginning of the simulation period, all five trucks are randomly placed in a 350 meter length of the highway, such that no two trucks overlap each other. At the end of the 5 minutes simulation period, all trucks must be travelling in a platoon under the restrictions described above. Your flooding algorithm must correctly execute even when the network configurations of the trucks change. Each intermediate wireless node that receives the vehicle sensor data packet must flood it onto all other outgoing links based on the flooding algorithm above.

Measure and report the average throughput, packet loss and end-to-end latency of the transmission of sensor and control data packets that are forwarded by the flooding algorithm through the VANET. Compare and plot these performances for different experiments with different number of trucks in the VANET.

5 What to do and turn in

After you have implemented, debugged, and thoroughly verified your truck platooning and flooding algorithm implementation, submit the following in Canvas on or before the due date.

a. Complete truck platooning simulation application codes and the flooding algorithm codes (which should include comments for full credit).
b. An execution trace of some important nodes (of important events only (packets received and sent), just enough to demonstrate the flooding algorithm and the truck platooning application works correctly.
c. A brief (2 pages) report that describes your system architecture, design and implementation issues, including sections on introduction and motivation, problem statement, details of your algorithms, experimental performance results, and conclusions. The experimental performance results must include the graph of the average throughput, packet loss and end-to-end latency of the transmission of sensor and control data packets for different number of trucks in the VANET. *Describe clearly and in detail the techniques and algorithm that you use for initiating truck platooning and maintaining the platoon configuration.*

You will demonstrate your program to the TA or me on March 21, 2014 in Davis 123 on the COE network of Linux computers.

6 Research Projects

Students may also choose to complete the following research projects in lieu of the above project. You can choose to complete only some selected topics from the list of research problems below. You have greater leverage in defining the actual problems and solutions to the problem as you analyze the systems and formulate your own problem. Some research problems may be further divided into several sub-problems that can be completed as different class projects. In each of the research problem, there are requirements for designing and implementing the algorithms and protocols, demonstration of the proper working of the system and performance measurements and evaluation of the system.

6.1 Heavy Truck Cooperative Adaptive Cruise Control (CACC)

A CACC system enables vehicles to electronically link, forming close-following drafting pairs on approved stretches of highway. The technology uses DSRC (IEEE 802.11p) vehicle-to-vehicle wireless communications (for convenience, you may use IEEE 802.11n instead) to allow two trucks to safely follow each other at a shorter distance than unassisted drivers can manage, while the driver retains steering control. CACC has the potential to significantly reduce fuel use for both the trailing and leading trucks – road experiments have shown 13% and 11% respectively. Initial analyses indicate that this could translate to fleet savings on the order of $8,000 in fuel/year/truck.

The following are some the research problems that students can choose for their class projects. You can consult with me to define the problems more specifically or provide the analysis of the problem yourselves and derive the solutions to your own set of problems.

1. Managing communication clusters of trucks dynamically. Derive the method for managing communication between trucks as they move in and out of range for their wireless and mobile ad-hoc networks.
2. Improving communication reliability between trucks in the VANET.
3. Improving communication by reducing congestion, interference and channel fading in the vehicle-to-vehicle communication.
4. Improving security and authentication methods in the VANET.
5. Improving efficiency of broadcast communication.

6.2 Network Assisted Navigation/GPS

The main objectives are to (1) optimally distribute navigation/GPS data across existing ad hoc networks, and (2) use distributed data to improve situational awareness.

The first challenge is to distribute navigation/GPS data over robust wireless ad hoc network architecture that frequently has limited connectivity. The algorithms for disseminating navigation/GPS data must be implemented with minimal bandwidth requirements. The second challenge is to distribute AGPS (Assisted GPS) and DGPS (Differential GPS) data. AGPS data is required for rapid acquisition and re-acquisition of GPS data in order to reduce time to first/subsequent fix. DGPS data is required for position accuracy improvement (reduce relative/global position error). The architecture should be designed for future expansion including RF ranging, smart phones, etc.

The following are some the research problems that students can choose for their class projects. You can consult with me to define the problems more specifically or provide the analysis of the problem yourselves and derive the solutions to your own set of problems.

1. Develop and implement distributed algorithms for efficient dissemination of AGPS and DPS data for wireless and mobile ad hoc networks.
2. Implement and evaluate the performance of ad hoc network routing protocols based on OLSR (Optimized Link State Routing protocol).
3. Implement and evaluate efficient multicast algorithms for AGPS and DGPS data.
4. Improving communication reliability for data dissemination in MANETs.

5. Improving communication throughput by reducing congestion, interference and channel fading in MANETs.
6. Improving security and authentication methods in MANETs.