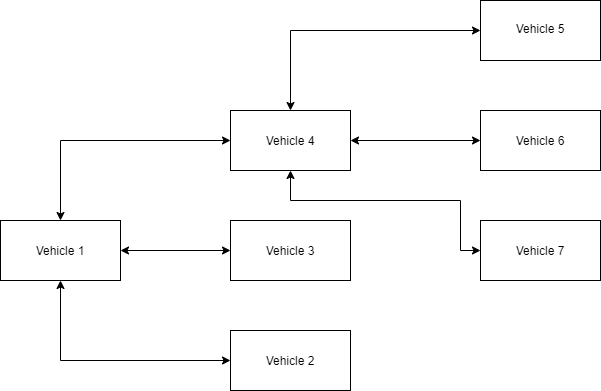
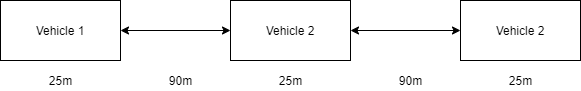
**Truck Platooning Use case**

Distributed Architecture: Distributed Component Architecture. The leading vehicles communicate with at most three vehicles behind.



The length of a truck in Germany is between 18 and 25 meters. The following distance between two trucks should be equivalent to the distance travelled by the vehicle in 4 seconds [1]. The maximum speed of trucks is 80 km/h. So, the recommended distance between two trucks is about 90 meters. For a convoy of three, the total distance between the leading truck and the last truck is 205 m. If the convoy de-couples to allow another vehicle in, the distance between the trucks is increased by 205 m. I a convoy of 4 with each vehicle decoupled, the total distance is 665 m



Parallel Architecture: The memory architecture for each multicore processor in each vehicle is a distributed memory architecture. Increases redundancy due to the high risk of the task at hand. The different functions of a vehicle have a processor assigned to that function.

Scalability. The number of vehicles in the convoy should be able to vary between 2 and 20

Latency. The latency of the communication between the vehicle in the front of the convoy and the last vehicle in the convoy should be low to reduce the possibility of accidents.

Synchronized Clock: The clocks in each vehicle are synchronized using GPS by Simple Network Time Protocol.

For the network the requirements are high bandwidth, low latency, long range, fault tolerance and deterministic characteristics. The network chosen is Long range Wi-Fi.

Mission: Develop a truck platooning system, to reduce fuel consumption and carbon dioxide emissions by 20 %.

The platooning concept can be defined as a collection of vehicles that travel together, actively coordinated in formation. Some expected advantages of platooning include increased fuel and traffic efficiency, safety, and driver comfort. [2]

The vehicles behind (trucks and passenger cars) follow the lead vehicle automatically; both laterally and longitudinally. Vehicles may join or leave the platoon dynamically e.g. leave on arrival at the desired destination.[3]

**Platoon Communication Protocol**

* The idea of the protocol is to establish a dedicated CAN bus for platooning, let us call it CANPlatoon. This CANPlatoon is integrated part of all existed CANs in the vehicle.
* CANPlaton reads/ and writes all relevant data for the platooning task.
* CANPlatoon allows other leading of following vehicle to interconnect.
* This is not possible without V2V gateway, which is a wireless medium for allowing vehicles to reach CANPlatton of another vehicle.
* Each vehicle is equipped of 2 V2V module to allow to vehicles to interconnect simultaneously.
* V2V module is responsible to interconnecting vehicle and to forward and receive defined messages (some data to other vehicle and get data from other vehicles) however not all received messages can pass.

Data on the platoon system includes (modifications can take place in further project steps)

* Leading truck current velocity and heading
* Leading truck waypoints (GPS points, desired velocity, yaw angle)
* Shared Data structure between all trucks in Platoon (example of it)

|  |  |  |  |
| --- | --- | --- | --- |
| Info/vehicle ID | 0 | 1 | 2 |
| Sensors status | Sensor.radar, sensor.speedometer … | … | … |
| Vehicle ID ‘unique’ |  |  |  |
| Lead truck | 1’lead truck’ | 0 ‘not lead truck’ | 0 ‘not lead truck’ |
| Slave truck | 0 | 1 | 1 |
| Vehicle order in Platoon | 0 ‘lead’ | 1 | 2 |
| Request to join |  |  |  |
| Request to leave |  |  |  |
| Current speed |  |  |  |
| Time stamp |  |  |  |
| Leaving |  |  |  |
| Left |  |  |  |
| Joining |  |  |  |
| Joined |  |  |  |
| And etc |  |  |  |

This data structure is written by all participating vehicles and every vehicle updates its values and has its own copy

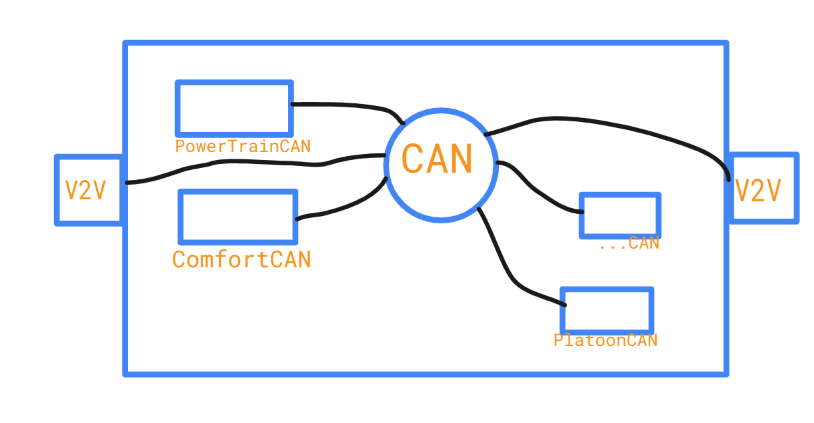


Figure 1 CANPlatoon Communication Protocol

**How our platoon system works?**

Any autonomous or semiautonomous vehicle there are 2 controls algorithm the first one is longitudinal control and lateral control.

Lateral control algorithm is executed exclusively in each vehicle. However, data form other vehicles help in lateral control algorithm.

After a vehicle joins the platoon, its longitudinal algorithm keeps fixed set speed and so fixed distance to the front vehicle, however additional inputs come from leading vehicle. The data is throttle and brake, these data comes from the leading vehicle. This makes the behaviour of the system proactive instead of responsive. In case of a vehicle fails to keep distance; its speed is lower than set point and the system cannot compensate, the leading vehicle adjust speed of the whole platoon to persevere platoon consistency and safety.

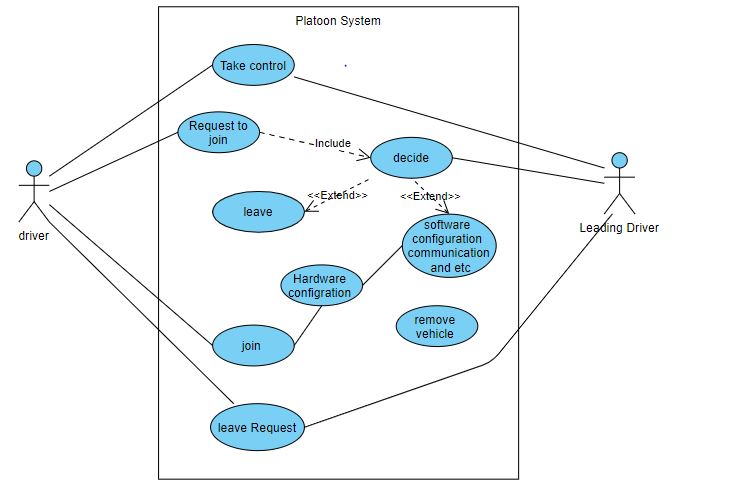
The following vehicles automatically strive to maintain the specified gap to the vehicle in front and the path and trajectory as specified by the lead vehicle. The local systems in the following vehicle can also take over in emergency situations and during loss of communication [3].

In case of communication failure. The system alerts the driver to be more ready for taking control. Lateral control works fine missed information from the leading vehicle will not cause any serious degradation on lateral control.

Longitudinal system will experience a degradation since throttle and acceleration commands will not reach following vehicles, the results are responsive response lag of maintain the required distance.

A radar system is responsible for measuring the distance with front vehicle, the platoon system in each vehicle sets the proper distance based on the speed that platoon drives, higher speed more distance is needed. ISO26262 guides automobile manufacturers in specifying distance to front vehicle.

**Use Case**



Programming each processor using OpenMP for multithreading and CUDA. The hardware chosen is the Nvidia Drive PX2 computer.

[1] <https://www.cedr.eu/download/Publications/2010/e_Distance_between_vehicles.pdf>

[2] Communication Protocol for Platoon of Electric Vehicles in Mixed Traffic Scenarios

[3] OVERVIEW OF PLATOONING SYSTEMS