Faculty of Engineering & Applied Science

ENGR 4941U
Capstone System Design II

Design of Smart Traffic Signals for Autonomous Transportation and Connected Communities in Smart Cities

Revised R1: Detail Design and Integration Testing Report

Group#: 8
Section CRN#:73459
Advisor: Dr. Hossam Gaber
Coordinator: Dr. Vijay Sood
Due Date: ?, 2023

First Name	Last Name	Student Number
Tirth	Patel	100751761
Vatsal	Patel	100728903
Saro	Karimi	100755079

Design Architecture & Key Functions

Design Architecture

In the Figure 1.5, we made a simple architecture of the smart traffic signal. Every car is connected to the edge server via wifi and sends their GPS coordinates. The cars location will be sent to the raspberry pi and the edge server will make some changes based on the volume of the cars and their coordinates. First, the coordinates will be used to determine the position of the cars and which direction and side of the road they are presented. Also, we can find out presence of the cars too. By having the wifi connection and assigning an ip address to each car we could easily find the number of cars passing or waiting at the intersection. For more improvement, we would consider using a camera to detect the pedestrians for the newest scenarios we made. For example, this would help to increase the timing of the pedesterians passing if they need more time to pass.

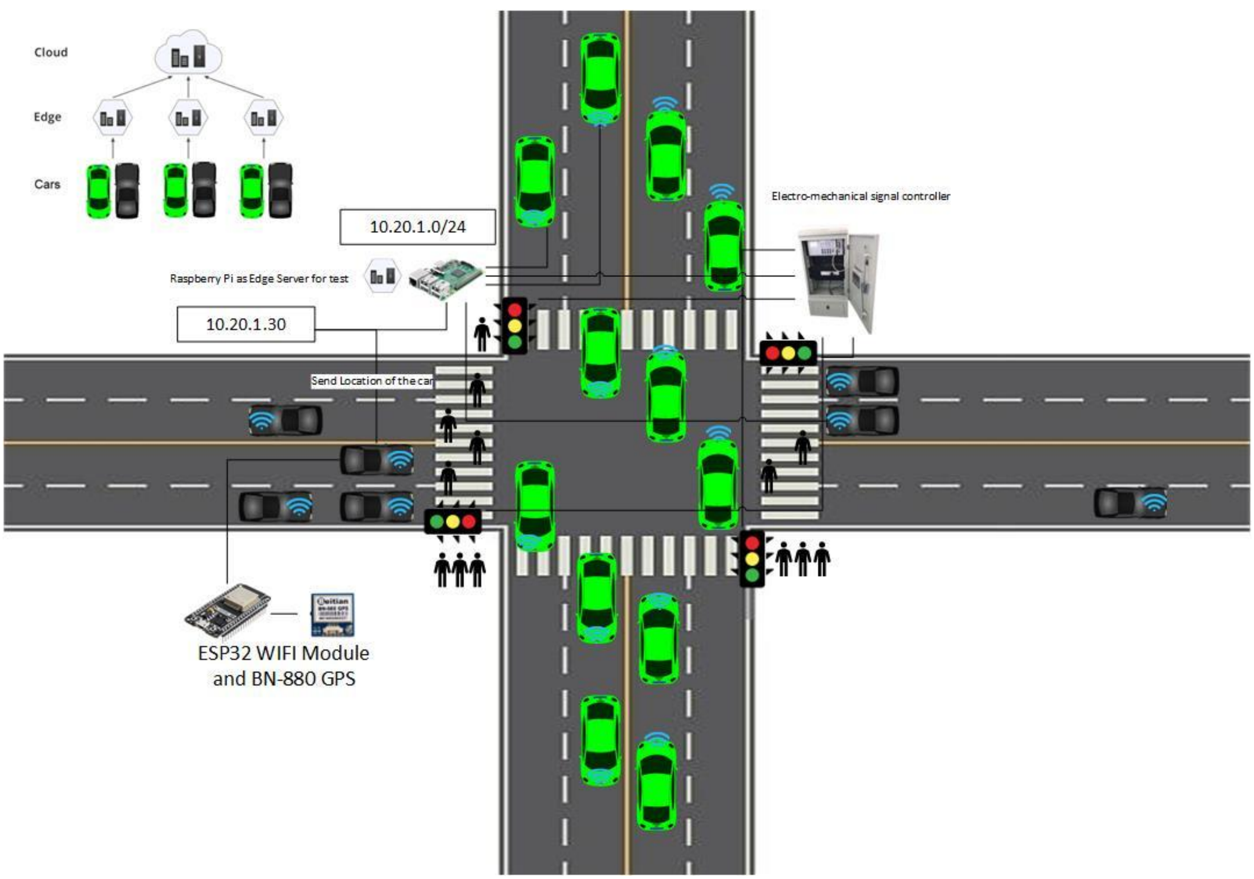

Figure 1.5: Design Architecture

Key Functions:

1. Collect and analyze real-time data from the cameras, sensors and GPS devices. This allows them to adjust signal timings based on current traffic conditions, such as congestion, accidents, and weather.
2. Adjusts their timing based on the volume and speed of vehicles approaching the intersection. This can help reduce wait times and improve traffic flow.
3. Gives priority to public transit vehicles and emergency vehicles, allowing them to move through intersections more quickly and efficiently.
4. Multiple traffic lights can be inter-connected to reduce stop-wait conditions. For example, the controller can gather data about how many cars enter the street at one position and how many cars exit at one position and can dynamically adjust multiple traffic lights to reduce congestion.

Test Case Scenarios

Scenario1:

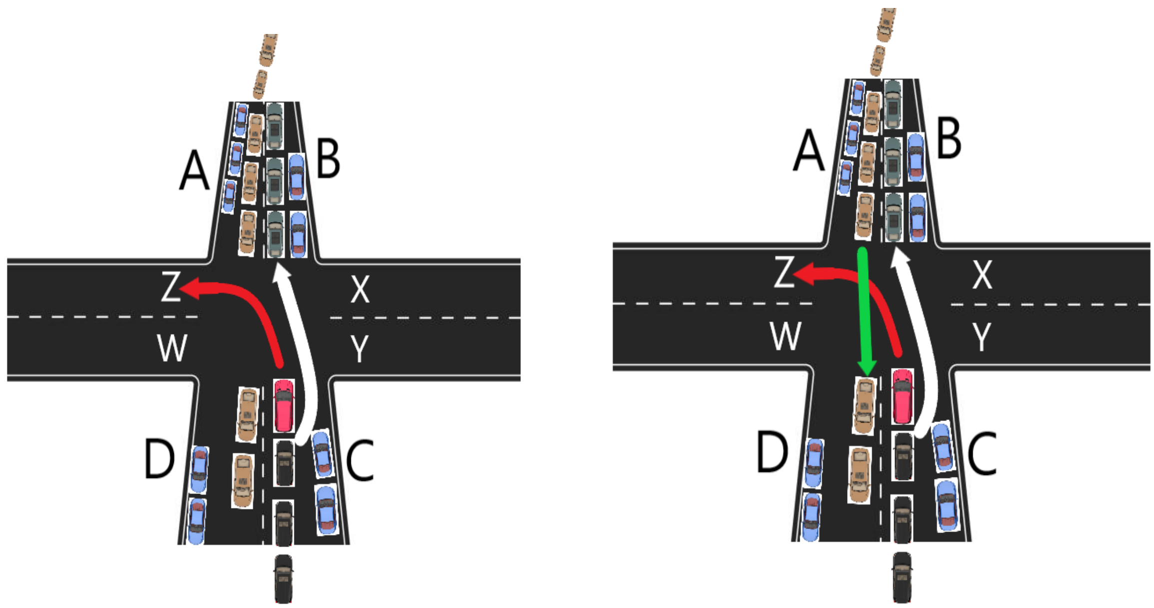

Figure 1.6 & 1.7: Test Case Scenario 1

In this scenario, there is traffic congestions because of the red car that is waiting for the opposing traffic to make the left turn. This is a very common case of traffic congestion that we can see at the intersection where there is no designated left turning lane. The Raspberry PI will recognize the red car is trying to turn left because it has been stopped for a certain period of time. In response, it will turn all the light at road A to red for a few seconds allowing the red car to safely turn left and the signal at road A will return to green.

Measurements:

- This scenario is based on a real experience of one of the team members at rush hours for the intersections in between Major Mackenzie to Crosby Avenue at Young street. He has experienced a huge difference in wait time cause by the red car situation which added around 5-10 minutes of wait time to this 1 km street that takes 1 minute to pass in normal hours.
- Consider the speed of the yellow cars are 50 km/hour and there are more of them tries to pass the intersection so they could not stop for the red car to pass since it will cause accident. Also, if we would consider the speed to be less than 50 km/hour, the possibility of the yellow cars to stop for the red car is almost zero.

- The other side of the intersection which is not a busy intersection will not turn green until there is a car or pedestrian tries to pass which even in this case takes at least 2 minutes after the detection of the car or the pedestrian.
- The black cars are adding up and if the red car don't move within 15-30 seconds, the wait on this intersection will effect the other intersections which the black cars are coming from and that will cause a major traffic.
- By our measurments and considerations, we would assume that the best option to solve this problem would be for the yellow cars to stop for 5 seconds so the red car could pass. This may take around a 1 minute process considering the traffic signal transformation, and the pedesterians, but it will save more time compared to 10 minute wait time our team member faces everyday. So this solution could eventually save around 5 minutes of wait time and makes the traffic smoother and the drivers less likely to get angery.

Scenario 2:

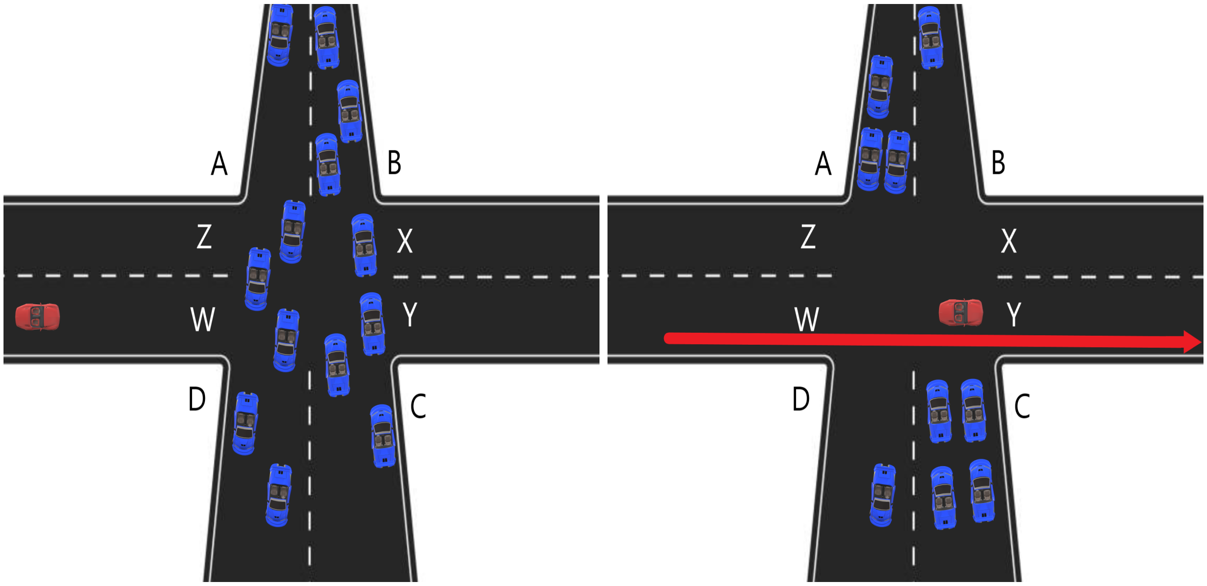

Figure 1.8 & 1.9: Test Case Scenario 2

In this scenario, the roads W,X,Y,Z are not busy whereas roads A,B,C,D are busy as it is the main road. Our system is able to update the traffic lights based on the information gathered in real time. What it means is that if there is no traffic roads W,X,Y,Z the light will remain red. If our system detects an incoming car the signal at roads A,B,C,D will turn red and signal at roads W,X,Y,Z will turn green. Furthermore, if our system detects that there is no incoming traffic on roads W,X,Y,Z then it will turn the signal to red earlier than the usual timer.

Measurements:

- The blue cars are assumed to be on a busy major street. The wait time for the light to turn green is 5 minutes but we can decrease this by finding out the volume of the cars in the major street.
- So assume that there are a volume of 20 cars trying to pass the intersection when the red car is not there. This volume of cars are normal for this major street in a rush hour. But sometimes the volume can drop to 5 cars and lower.
- By finding out the geo location of the blue cars and transmitting their data to our edge server we can find out how many cars are there and decide base on that.
- So assume there are 5 cars in the major street and the red car tries to pass. So in this case since the ratio of the (available cars:Volume of cars in a busy hour) is 1:4 then we will turn the light green for the red car so that it will pass.
- This helps since if we turn the light green based on the preset timing and there are 1:1 ratio of cars, we will cause traffic, but by considering the 1:4 ratio and try to pass the red

car before the ratio returns to 1:1, we could avoid wait time and massive traffic for the blue car side.

Validation of measurements: write down how we got the measurements.

Scenario 3:

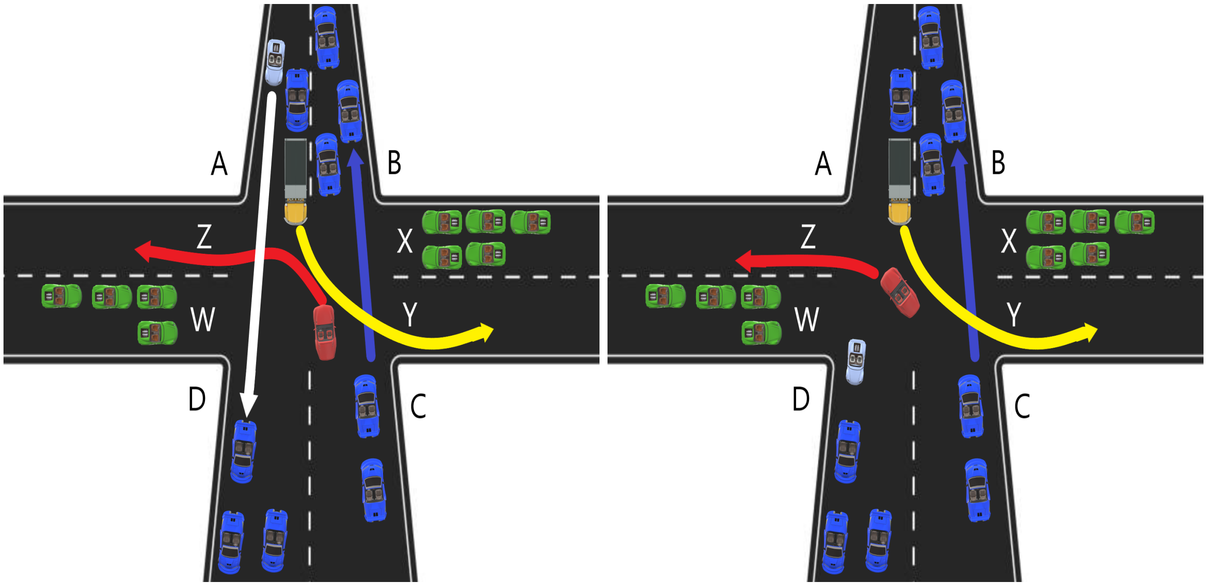

Figure 1.10 & 1.11: Test Case Scenario 3

In this scenario, we have a red car that is turning left on to road Z and yellow truck that is turning left on to road Y. The red car is having difficult seeing the incoming traffic because the yellow truck is blocking the line of sight. Our system will have a caution light which will light up if the system detects the the incoming traffic from road A so that the red car can turn left in a safe manner.

Measurements:

- The traffic flow the blue side is not stopping since it is really busy and the yellow truck could not pass the steet in good amount of time.
- The size of the yellow truck is bigger than the red car, so by the laws of physics the driver in the red car could not see the white cars side.
- From one of our sources in the previous reports, we found out that, there is 58% chance of T-boning in this situation and by their analysis in USA, 40% of 5.8 million police-reported vehicle crashes has been considered drivers t-boning each other in a left turn situation like this.
- We propose to lower this huge number only by installing a caution light. If the caution light turns red means it is not safe at all for the red car to do the left turn. If it turns green it means there are no danger for the red car to do the left turn. Also, we could say that for the yellow truck too since it is doing a left turn. So, this light will be visible to the driver.
- The light will decide the choice based on the white car's location using the GPS coordinates we got from it. So, we can do some calculations.

- Assume the white car is coming toward the light at the speed of 60 km/hour. The average speed the red car needs to travel for doing the left turn is 10 km/hour. By the guidelines of Toronto, the maximum length of an intersection is around 30 meters. What is the safest range that the white car should be away from the intersection so the red car could pass it?
- Answer:
    1. Find the diagonal of the intersection: a = 30 meter by a=30 meter square => the diagonal of the square = a*sqrt(2) = 30 *sqrt(2) = approximately 45 meters of traveling is needed for the red car.
    2. Time is needed for the red car to pass through the intersection: we assume that the speed is consistent and equal to 10 km/hour = 2.78 meters/second. Time = Distance/Speed => Time = 30 (meters)/ 2.78 (meters/second) = 10.8 seconds is needed for the red car to travel.
    3. Danger zone 'x' of the red car: if the distance of the white car is less than x then the red car cannot do a left turn since it is in the danger zone. x = time * speed => speed = 60 km/hour = 16.67 meters/second => x = 10.8 (seconds) * 16.67 (meters/second) = 180 meters => x = 180 meters
- We found out that if the white car or cars are closer than 180 meters to the intersection then the red car cannot do a left turn. But since we want to consider human error and careless driving of some people driving faster than the speed limit, the optimal range will be 285 meters and more. We can optimize the range by knowing the speed of the car coming toward the intersection if we could somehow get the speed of the car to our ESP32 module so it could send it to the edge server for more analysis.

We need to get a flow chart for this, to get a clear explanation.